

Ministère de l'Enseignement Supérieur de la Recherche et de l'Innovation

Université Assane Seck de Ziguinchor

UFR des Sciences Economiques et Sociales



Mémoire de fin d'études

Pour l'obtention du diplôme de Master

Mention : : Management des Systèmes d'information (MSI)

Spécialité : Management des Systèmes d'Informations Automatisés (MSIA)

**SUJET : Mise en place d'une application web pour la gestion d'élections de la
Commission Nationale des Élections (CNE) de la Guinée Bissau**

Présenté par : Mr Aliu Djaló

Soutenance le 28/03/2022

Sous la Direction du : Dr Papa Alioune Cissé

Devant le jury ci-après :

Prénom (s) et nom	Grade	Jury	Etablissement
Pr. Melyan MENDY	Professeur	Président du jury	UASZ
Dr. Mouhamadou GAYE	Maître de conférences assimilé	Examineur	UASZ
Dr. Abel DIATTA	Maître de conférences assimilé	Rapporteur	UASZ
Dr Papa Alioune CISSE	Maître de conférences assimilé	Encadrant	UASZ

Année Académique 2019-2020

Résumé

Le but de ce projet est d'étudier le système d'informations de la Commission National des Élections de la Guinée-Bissau afin de l'informatiser.

Il s'agit donc de mettre en place une plateforme web permettant la gestion des élections pour la Commission Nationale des Élections en Guinée-Bissau. Cette plateforme web est développée avec les Framework Laravel et Bootstrap et utilise le système de gestion de base de données MySQL.

Elle permettra de gérer les élections de façon plus simple, plus efficace et sécurisée. En effet, la plateforme permet de gérer les élections, l'insertion des résultats de bureaux de vote, de générer automatiquement les rapports et en fin les statistiques nécessaires du scrutin.

Abstract

The purpose of this project is to study the information system of the National Election Commission in order to computerize it.

It is therefore a question of setting up a web platform allowing the management of elections for the National Election Commission in Guinea-Bissau. This web platform is developed with the Laravel and Bootstrap Frameworks and uses the MySQL database management system.

It will make it possible to manage elections in a simpler, more efficient and secure way. Indeed, the platform makes it possible to manage the elections, the insertion of the results of polling stations, to automatically generate the reports and in the end the necessary statistics of the ballot.

Dédicace

A mes très chers parents

A mes très chers grands frères Tidjane Djaló, Maimuna Djaló et Aissato Djaló,

A mon tuteur Bourama Seydi,

A mon très cher meilleur ami, Cheikhou Diokou,

A mes amis d'enfance et l'Association des Bissau Guinéens.

Remerciements

Je rends grâce à Allah, le Tout-Puissant qui, par sa grâce, m'a permis d'arriver où j'en suis aujourd'hui, preuve une fois de plus de sa gratitude, de Sa Miséricorde envers moi. Je prie aussi sur le Saint prophète (PSL).

Je remercie particulièrement mon encadrant, le Dr Papa Alioune Cissé pour sa disponibilité, pour la rigueur dont il a faite preuve dans le travail, me poussant ainsi à donner le meilleur de moi-même, également pour tout le savoir transmis.

Je remercie les membres du jury qui ont bien voulu participer à ma soutenance.

Je remercie tout le corps professoral du Master Management des Systèmes d'Informations Automatisés (MSIA) de l'UFR Sciences Économiques et Sociales de l'Université Assane Seck de Ziguinchor pour ces années de formation et tout le peuple du Sénégal.

Je remercie le personnel de la Commission National des Élections de la Guinée-Bissau pour leurs collaborations durant la réalisation de ce mémoire.

Je remercie tout le personnel du consulat de la Guinée-Bissau à Ziguinchor pour leur soutien durant ces années de formation.

Je remercie tous mes camarades de promotion pour ces belles années de partage et d'émotions et en particulier Cheikhou Diokou.

Que ce travail soit le symbole de notre amitié dont vous avez toujours fait preuve.

Je remercie tout le personnel de l'université Assane Seck de Ziguinchor.

Je remercie aussi tous mes amis pour la chaleur humaine et le soutien qu'ils m'ont apporté.

Aussi à toutes les personnes qui m'ont aidé de près ou de loin à la réalisation de ce mémoire

Table des matières

Résumé	1
Abstract.....	1
Dédicace	2
Remerciements	3
Table des matières.....	4
Liste de Figure.....	8
INTRODUCTION GENERALE.....	11
Chapitre 1 : ETUDE DE LA GESTION DES ELECTIONS	12
Introduction	12
La gestion actuelle des élections	12
Avant les élections	12
Après les élections	12
Problématique de la gestion électorale actuelle.....	13
Problème de cas de retard et l'efficacité de la gestion électorale	13
Problème sur la restriction de l'accès aux informations de vote	14
Problème sur la sécurité des informations de vote	14
Problème de production des rapports de l'évaluation de vote	14
La solution aux problèmes de gestion des élections.....	14
Objectif et méthodologie	15
a. Objectif	15
b. Méthodologie	15
Conclusion.....	15
Chapitre 2 : ANALYSE ET CONCEPTION.....	16

Introduction	16
1. Outils d'analyse et de conception.....	16
a. Merise	17
b. UML	18
c. Choix de la méthode	18
2. Analyse du système.....	19
a. Description du système : rôle des acteurs	19
b. Diagrammes de cas d'utilisation	20
3. Conception du système	21
a. Diagramme de classe	21
b. Architecture logicielle.....	22
c. Conclusion.....	23
Chapitre 3 : ETUDE DES OUTILS TECHNOLOGIQUES	24
Introduction	24
1. Langages de programmation	24
a. Java Enterprise Edition (Java EE).....	24
b. PHP.....	25
c. DotNET	26
2. Choix de langage	27
3. Serveur d'application.....	28
Qu'est-ce qu'un serveur Web ?	28
Qu'est-ce qu'Apache ?	28
Comment fonctionne le serveur web Apache ?.....	29
Apache vs autres serveurs web.....	29

Les avantages et inconvénients d'Apache	30
4. Outils d'optimisation : Les Frameworks	31
a. Laravel	31
b. Boostrap	32
5. Bibliothèques.....	32
a. JQuery.....	32
6. Outils techniques : choix du SGBDR	33
1- MySQL	34
2- PostgreSQL.....	34
3-Oracle	35
4- Microsoft SQL Serveur	36
6- Tableau comparatif des SGBD	36
7. Outils supplémentaires.....	37
1. Visual Studio Code	37
2. WAMP Server.....	38
Conclusion.....	38
Chapitre 4 : IMPLEMENTATION ET PRESENTATION DE L'APPLICATION	39
Introduction	39
4.1 Architecture applicative 3-tiers de l'application web	39
4.2 Structure de la base de données.....	40
4.3 Structure du projet.....	40
4.4 Présentation de l'application	41
Profil Administrateur	42
Profil Chef Département de Statistique et Informatique	48

Profil de Président de la Commission Régional des Élections	52
Profil de Délégué	55
4.5 Conclusion.....	59
CONCLUSION ET PERSPECTIVES	60
1. Conclusion.....	60
2. Perspectives.....	60
BIBLIOGRAPHIE ET WEBOGRAPHIE	62
ANNEXES.....	63
Annexe 1 : Fichier de configuration de la connexion .ENV	63
Annexe 2: Fichier de configuration de la base de données (database.php).....	64
Annexe 3 : Fichier composer.json	65
Annexe 4 : Dossier de fichier des migrations de base de données	65
Annexe 5 : Les Models et les contrôleurs	66
Annexe 6 : Les routes de l'application (web.php).....	67
Annexe 7 : Schéma de la base de données.....	69
Annexe 8 : La Commission Nationale Des Élections	74

Liste de Figure

Figure 1-Abstraction pour la conception de SI [2]	17
Figure 2 - Diagramme de cas d'utilisation	20
Figure 3 - Diagramme de Classes	21
Figure 4- L'Architecture Logicielle	22
Figure 5 - L'Architecture Applicative	39
Figure 6 - Structure de la Base de Données	40
Figure 7-Structure de projet	40
Figure 8-Page de connexion utilisateur	42
Figure 9-Page accueil administrateur	42
Figure 10-Menu localités	43
Figure 11-Bureaux de votes	43
Figure 12- Ajout de bureau de vote	44
Figure 13-Ajout de bureau avec succès	44
Figure 14- Menu gestion élection	45
Figure 15- Elections	45
Figure 16-Ajout d'élection	45
Figure 17- Election ajouté avec succès	46
Figure 18- Modification des informations d'élection	46
Figure 19- Gestion des utilisateurs	47
Figure 20-Ajout d'un nouvel utilisateur	47
Figure 21-Ajout réussi d'un nouvel utilisateur	48
Figure 22- Page accueil chef département	48
Figure 23-Dashboard National	49

Figure 24-Page d'accueil pour gestion des rapports d'élection	50
Figure 25-Rapport national de résultats électoraux	50
Figure 26-Page menu rapports régionaux	51
Figure 27-Rapport région de Gabú.....	51
Figure 28-Page Accueil Président de la Commission Régional D'Elect ion	52
Figure 29-Page Accueil Gestion de Rapports Régionaux	53
Figure 30-Page gestion rapport cercle.....	53
Figure 31-Rapport de Résultats Electoraux de Cercle 15	54
Figure 32-Page Accueil gestion rapport Secteurs à la CRE	54
Figure 33- Page accueil Délégué	55
Figure 34-Visualisation sectorielle de résultats	55
Figure 35-Page saisie de résultat de vote	56
Figure 36-After insertion de résultat	57
Figure 37-Statistique Sectoriel d'Elect ion	57
Figure 38-Menu Gestion de rapports de secteur	58
Figure 39-Rapport sectoriel résumé de résultat	58
Figure 40-Rapport résultat de bureaux de vote.....	59
Figure 41-Datawarehouse	61

Liste des Tableaux

Tableau 1- Choix de la méthode	18
Tableau 2 – Tableau comparatif des plateformes de développement d’applications web	27
Tableau 3 - Tableau comparatif des SGBD	37

Liste des acronymes

AJAX : Asynchronous JavaScript and XML

ANP : Assemblée Nationale Populaire

API : Application Programming Interface ou Interface de Programmation d’Application en Français

BDD : Base de Données

CGI : Common Gateway Interface

CLR : Common Language Runtime

CNE : Commission Nationale d’Elections

CRE : Commission Régionale D’Election

CSS: Cascading Style Sheet

HTML: Hypertext Markup Language

IDE : Integrated Development Environment ou Environnement de développement intégré

Merise : Méthode d’étude et de réalisation informatique pour les systèmes d’entreprise

MSIA : Management de Systèmes d’Informations Automatisés

MVC: Model View Controller

OLAP: Online Application Protocol

OMG : Object Management Group

PDF : Portable Document Format

PHP : Personal Home Page, puis Hypertext Preprocessor

SDK : Software Development Kit

SGBD : Système de Gestion de Base de Données

SQL: Structure Query Language

UML: Unified Modeling Language

USB: Universal Serial Bus

XML: Extensible Markup Language

INTRODUCTION GENERALE

La Commission Nationale des Élections de la Guinée Bissau, désigné **CNE**, est un organe indépendant et permanent qui collabore avec l'Assemblée Nationale Populaire et est responsable de l'organisation et la gestion du processus électoral et référendaire de la Guinée-Bissau. La CNE est l'unique organe pour organiser et gérer les Élections Présidentielles, Législatives et Autarchiques.

Toutefois, sa gestion des élections est jusqu'à présent assurée par des fichiers Excel imprimés et que c'est seulement au niveau des régions qu'est utilisé un ordinateur qui n'est pas connecté à internet et qui sert juste à concevoir et imprimer un rapport régional et le reste du travail est fait manuellement.

C'est ainsi qu'une politique de modernisation de la gestion des élections a été initiée avec la mise en place d'une plateforme web qui permettra d'avoir une gestion collégiale, efficace et sécurisée ; un suivi des élections et la production de rapports et statistiques fiables.

Pour bien réaliser ce travail, nous l'avons divisé en quatre chapitres :

Le premier est consacré à l'étude du système d'informations actuel de la CNE afin de connaître ses limites. Ainsi, face à ses limites, nous proposons une solution, des objectifs et une méthodologie pour les atteindre.

Le deuxième chapitre concerne la modélisation et la conception d'un nouveau système de gestion des élections pour la CNE en vue de concrétiser la solution proposée.

Dans troisième chapitre, il sera question d'étudier les outils technologiques que nous avons utilisés afin de justifier notre choix sur ces derniers contrairement à d'autres outils existants.

Le dernier chapitre concerne la mise en œuvre de la solution proposée c'est-à-dire l'implémentation et la présentation de l'application.

Chapitre 1 : ETUDE DE LA GESTION DES ELECTIONS

Introduction

Dans ce chapitre, il est question d'étudier en premier lieu, la gestion actuelle des élections de la Commission Nationale des Élections. En second lieu, nous énumérerons ses problèmes pour savoir ses besoins afin de fixer les objectifs à atteindre et la méthodologie à adopter pour y parvenir.

La gestion actuelle des élections

Actuellement, la CNE gère les élections en Guinée Bissau de façon manuelle. En effet, certains rapports comme ceux produits dans les cercles, les secteurs électoraux sont établis manuellement. En outre, le dépouillement national des résultats de votes est effectué en utilisant des fichiers Excel.

Pour expliquer la gestion actuelle des élections en Guinée Bissau, nous allons exposer le déroulement des différentes tâches avant et après élections.

Avant les élections

La Commission Nationale d'Élections (CNE) organise le recensement électoral au niveau national et prépare tous les moyens nécessaires pour la réalisation des élections. Après, la CNE désigne pour chaque région une commission qui va organiser les élections au niveau régional.

La Commission Régionale des Élections (CRE) est chargée de préparer les moyens nécessaires et la formation des délégués qui vont aux secteurs qui composent chaque région.

Les délégués préparent les membres de bureaux de votes et poursuivent avec la distribution de matérielles nécessaires pour chaque groupe de membres de bureaux.

Les membres de bureaux de vote s'installent et le vote commence à 7 heures de matin le jour prévu pour les élections et se termine à 17 heures voir **Article 69 d'Annexe 8**.

Après les élections

Le président de bureau de vote poursuit avec les opérations préliminaires : l'ouverture des unes, le comptage et enfin l'élaboration manuelle du rapport des résultats de bureaux de votes qui sera transmis au délégué de la CRE.

Les délégués de la CRE rangent les rapports de résultat des bureaux en les mettant dans une enveloppe. Ensuite, ils parcourent des kilomètres pour transmettre cette enveloppe au niveau de la région.

Après cela, la CRE poursuit avec la saisie de rapports de résultat de bureaux en utilisant un ordinateur que n'a pas de connexion internet et qui sert juste à imprimer le rapport régional.

Le reste de travail c'est-à-dire le rapport par cercle électoral et secteur sont fait manuellement avec les calculatrices au niveau de la région un jour ou deux jours après les élections.

Ensuite les rapports régionaux sont photocopiés et remis à chaque représentant de candidat. Le président de la commission régionale prend une copie du rapport régional et l'ordinateur pour aller à la CNE. Une fois arrivé, l'informaticien au niveau de la CNE extrait les données sur les ordinateurs pour faire le dépouillement national avec Excel et présenter le résultat national avec PowerPoint quelques jours après les élections.

Problématique de la gestion électorale actuelle

L'étude du système électoral actuel de la CNE après les élections montre des problèmes qu'il convient d'exposer comme suit :

Problème de cas de retard et l'efficacité de la gestion électorale

Actuellement, le système utilisé par les délégués de CRE au niveau des secteurs est manuel. Toutes les données (résultats de bureaux) sont vérifiées aux secteurs avec des calculatrices. Après cela, les délégués de la CRE mettent les rapports (résultat des bureaux) dans une enveloppe et se réunissent à la région pour commencer l'introduction des données en utilisant un ordinateur, permettant uniquement l'impression du rapport régional. Puis, tout le reste du travail de la CRE est effectué manuellement avec des calculatrices.

Après l'impression des rapports régionaux, les ordinateurs sont rangés et les rapports imprimés sont mis dans une enveloppe afin de les transporter à la CNE pour faire l'évaluation nationale avec Excel puis présenter sur PowerPoint.

De ce fait, les résultats de bureaux sont transportés d'abord des lieux de vote vers les délégués, ensuite des délégués à la CRE et enfin, vers la CNE. C'est pour cela que les résultats arrivent très souvent en retard. Ce qui explique plusieurs cas de retard et aussi l'inefficacité du système.

Problème sur la restriction de l'accès aux informations de vote

Il convient de remarquer que les délégués de CRE ne produisent pas de rapport sectoriel car le système informatique utilisé sert juste à saisir les rapports de résultats électoraux de bureau et imprimer le rapport régional. En outre, même pour produire le rapport d'évaluation par cercle électoral, ils le font manuellement à la région au lieu de le faire au niveau du secteur. A cela s'ajoute le fait que les électeurs ne peuvent pas connaître les résultats du secteur électoral puisque les délégués se réunissent au niveau de la région. Ceci justifie encore les cas de retard, restriction de l'information et l'inefficacité du système.

Problème sur la sécurité des informations de vote

Les fiches ou les rapports sont gardés dans les ordinateurs ou dans des dispositifs de stockage amovibles (disque dur externe ou clé USB), dans leurs armoires en version papier.

Toutefois, ces ordinateurs ou ces dispositifs de stockage amovibles peuvent à tout moment être infectés par un virus qui soit capable de détruire ces fichiers. De même les risques incendiaires sont non négligeables.

Problème de production des rapports de l'évaluation de vote

Actuellement les rapports par cercle sont établis manuellement avec machine à calcul et il est plutôt facile de faire un rapport d'évaluation après une élection. Mais si ce rapport concerne les années passées, alors les produire reste très compliqué voire impossible car certains rapports de résultat de bureau peuvent être perdues. Ceci justifie encore une fois le retard lors de la production des rapports.

La solution aux problèmes de gestion des élections

Face à ces nombreux problèmes que rencontre le système électoral de la CNE, nous proposons comme solution une application web de gestion des élections pour la CNE.

Cette application assurera une gestion efficace et collégiale des élections, un accompagnement de l'introduction des données, un stockage sécurisé des informations de votes et une production automatique des rapports de résultat de bureaux juste après la saisie au niveau de terrain et statistiques de vote. Pour faire fin aux problèmes cité ci-dessus. Tous les rapports établis quelques jours après les élections désormais seront produisent quelques minutes après les élections.

Objectif et méthodologie

a. Objectif

La plateforme web que nous proposons comme solution, permettra d'atteindre les objectifs suivants :

- ✓ Gérer (ajouter, modifier et supprimer) les régions, cercles électoraux, districts électoraux, bureau de vote, les candidats et les élections ;
- ✓ Créer les utilisateurs ;
- ✓ Inscrire un candidat à l'élection ;
- ✓ Générer automatiquement les rapports par région, par cercle par secteur, ... ;
- ✓ Produire des statistiques générales, par région, par cercle, par secteur, ... ;
- ✓ Pouvoir imprimer les rapports, les statistiques, ...

b. Méthodologie

Pour atteindre ces objectifs, nous allons procéder comme suit :

- ✓ Concevoir un diagramme de cas d'utilisation pour montrer les acteurs du système et leurs atteintes ;
- ✓ Concevoir un diagramme de classes pour organiser le stockage des données ;
- ✓ Faire une architecture logicielle de la solution proposée ;
- ✓ Mettre en place un entrepôt de données (Datawarehouse) pour la production des statistiques fiables ;
- ✓ Choisir des outils technologiques à savoir :
 - Un langage de programmation ;
 - Un serveur d'application dans lequel l'application sera déployée ;
 - Des outils d'optimisation : Frameworks ;
 - Un SGBD pour stocker les données.
- ✓ Implémenter ces outils technologiques pour réaliser notre plateforme web ;
- ✓ Tester et Valider notre solution pour déterminer ses failles et manquements.

Conclusion

L'étude de la gestion élection actuelle de la CNE nous a permis de déceler un nombre important de problèmes auxquelles elle est confrontée.

Face à ces problèmes, nous avons proposé une solution et nous sommes fixés des objectifs à atteindre et une méthodologie pour concrétiser cette solution.

Chapitre 2 : ANALYSE ET CONCEPTION

Introduction

Cette partie concerne l'analyse et la conception d'un nouveau système informatique de gestion des élections pour la CNE qui permettra de concrétiser notre solution, donc de parvenir aux objectifs que nous nous sommes fixés.

Dans un premier temps, nous allons faire, une étude comparative entre des outils de conception (Merise et UML) afin de faire un choix sur l'un d'entre eux. Dans un deuxième temps, nous précéderons à la description du système en montrant ses différents acteurs avec leurs rôles. En dernier temps, nous parlerons de la conception proprement dite en faisant les diagrammes de cas d'utilisation et de classes, l'architecture logicielle du système et la mise en place d'un entrepôt de données.

1. Outils d'analyse et de conception

Un **Système d'Informations** (SI) est un système de gestion de l'information au sein d'une *organisation*¹. [2]

Un SI est en quelque sorte une extension de la mémoire humaine qui amplifie le pouvoir de mémorisation des acteurs de l'organisation et facilite leur prise de décision.

Objectifs d'un Système d'Information :

Permettre de collecter, structurer, stocker, traiter et diffuser l'information. Et aussi réduire le temps de traitement de l'information pour améliorer la proactivité, l'activité, la réactivité et le rendement d'une entreprise.

Un système d'information est un ensemble constitué d'éléments unis par des relations, ces éléments et ces relations étant munis de propriétés. La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place afin de pouvoir satisfaire au final les besoins des clients. Cette phase de conception nécessite une méthode d'analyse permettant de mettre en place un modèle sur lequel on va s'appuyer. En

¹ Une **Organisation** est une association de plusieurs personnes ayant des objectifs communs et bien déterminés.

outre, cette méthode permet de formaliser les étapes préliminaires du développement d'un système d'information afin de rendre ce développement plus facile à la réalité. [2]

Il existe plusieurs méthodes d'analyse. La partie suivante propose une étude de la méthode MERISE (Méthode d'Étude et de Réalisation Informatique pour les Système d'Entreprise) et du langage UML (langage de Modélisation Unifié) d'abord, et en dresser un tableau comparatif entre ces méthodes afin de choisir l'une d'entre elle pour la modélisation et la conception de notre système.

a. Merise

MERISE est une méthode systémique de conception des systèmes d'information (SI).



Merise propose un ensemble de formalismes et de règles destinées à modéliser² de manière indépendante les données et les traitements du système d'information.

Elle se propose pour les activités de conception, de développement et de réalisation de projets informatiques, avec comme objectifs la constitution d'un *système d'information*. Elle repose sur un ensemble de formalismes et de règles permettant de séparer les données de traitements, et sur la construction de modèles conceptuels, logiques et physiques de données, ordonnées au moyen du *modèle entité-association*. [2]

Cycle d'abstraction pour la conception de SI [2]

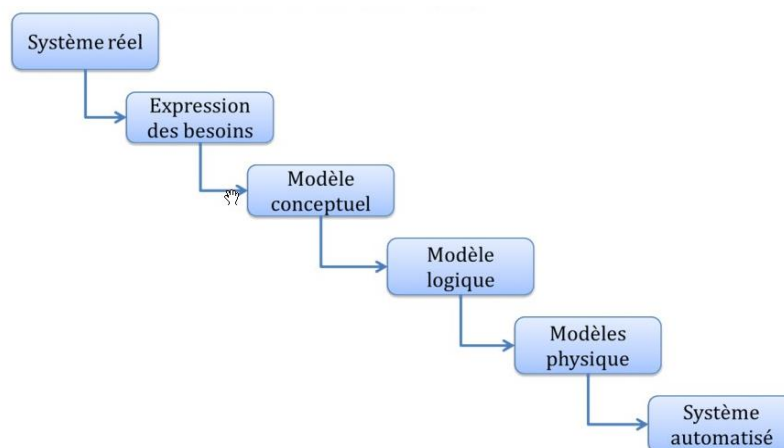


Figure 1-Abstraction pour la conception de SI [2]

² **Modéliser** consiste à donner une représentation structurée et simplifiée d'un système réel, afin de reproduire son fonctionnement sur ordinateur et de répondre de manière efficace et rapide à des questions que l'on pourrait se poser sur le système réel.

b. UML



UML est un langage de modélisation et de conception de systèmes informatiques, plus particulièrement adapté à la programmation orientée objet.

UML, normalisé par l’OMG (*Object Management Group*), est à la fois un langage de modélisation, un support de communication et un cadre méthodologique. C’est un langage formel défini par un méta modèle qui décrit de manière précise tous les éléments de modélisation (les concepts véhiculés et manipulés par le langage) et la sémantique de ces éléments (leur définition et le sens de leur utilisation).

UML se présente comme un métalangage par rapport aux langages de programmation, il repose sur une notation graphique indépendante des langages de programmation qui donne à chaque concept objet une représentation particulière, au moyen de neuf types de diagrammes qui sont autant de vues mutuellement cohérentes sur le même modèle :

- Les quatre diagrammes structures, qui donnent de l’application une vue statique :
 - Diagramme de classes,
 - Diagramme d’objets,
 - Diagramme de déploiement,
 - Diagramme de composants ;
- Les cinq diagrammes comportementaux qui donnent une vue dynamique de l’application :
 - Diagramme d’activités,
 - Diagramme de séquence,
 - Diagramme d’états-transition,
 - Diagramme de collaboration,
 - Diagramme de cas d’utilisation.

La confection de ces diagrammes au moyen d’un outil UML adéquat tel que *PowerAMC* permet de générer automatiquement le code qui correspond aux diagrammes, au choix en Java, C++, Visual Basic etc...

c. Choix de la méthode

Le tableau comparatif qui suit a permis de justifier notre choix sur UML

MERISE	UML
Méthode d’analyse et de conception de systèmes d’information	Langage de représentation d’un système d’information
Relationnel	Objet
Plus adapté à une approche théorique	Plus orienté vers la conception

Tableau 1- Choix de la méthode

D'après ce tableau comparatif, nous avons choisi UML comme outil de conception du fait qu'il s'agit d'un système de notation orienté objet et utilise un modèle **Top-down** c'est-à-dire qu'à partir du modèle conçu, on peut générer le code en Java, C++... [3]

2. Analyse du système

a. Description du système : rôle des acteurs

Dans notre système, plusieurs acteurs ayant des rôles différents interviennent.

Ces acteurs sont les suivants :

- ✓ **Administrateur du système** : il s'agit du profil chargé de gérer (créer, modifier, supprimer) les différents profils et utilisateurs du système, les régions, cercles électoraux, secteur, districts électoraux, les bureaux de votes et paramétrage du système...
- ✓ **Chef de Département de Statistiques et Informatique** : Il peut accéder à toutes les informations de scrutin voire les statistiques de toutes les régions et imprimer les rapports. Donc le profil chargé pour gérer le résultat national.
- ✓ **Président d'une CRE** : il s'agit du profil de responsable d'une CRE. Il peut aussi voir les statistiques et imprimer les rapports de scrutin de la région qu'il gère.
- ✓ **Délégué d'une CRE** : il s'agit du profil de responsable d'un secteur. Il joue le rôle d'utilisateur qui va insérer les actes d'apurement (résultat de bureaux) provenant de bureaux de votes. Il peut voir les statistiques et imprimer les rapports de scrutin de secteur ou cercle qu'il gère après l'insertion de résultats des bureaux.

b. Diagrammes de cas d'utilisation

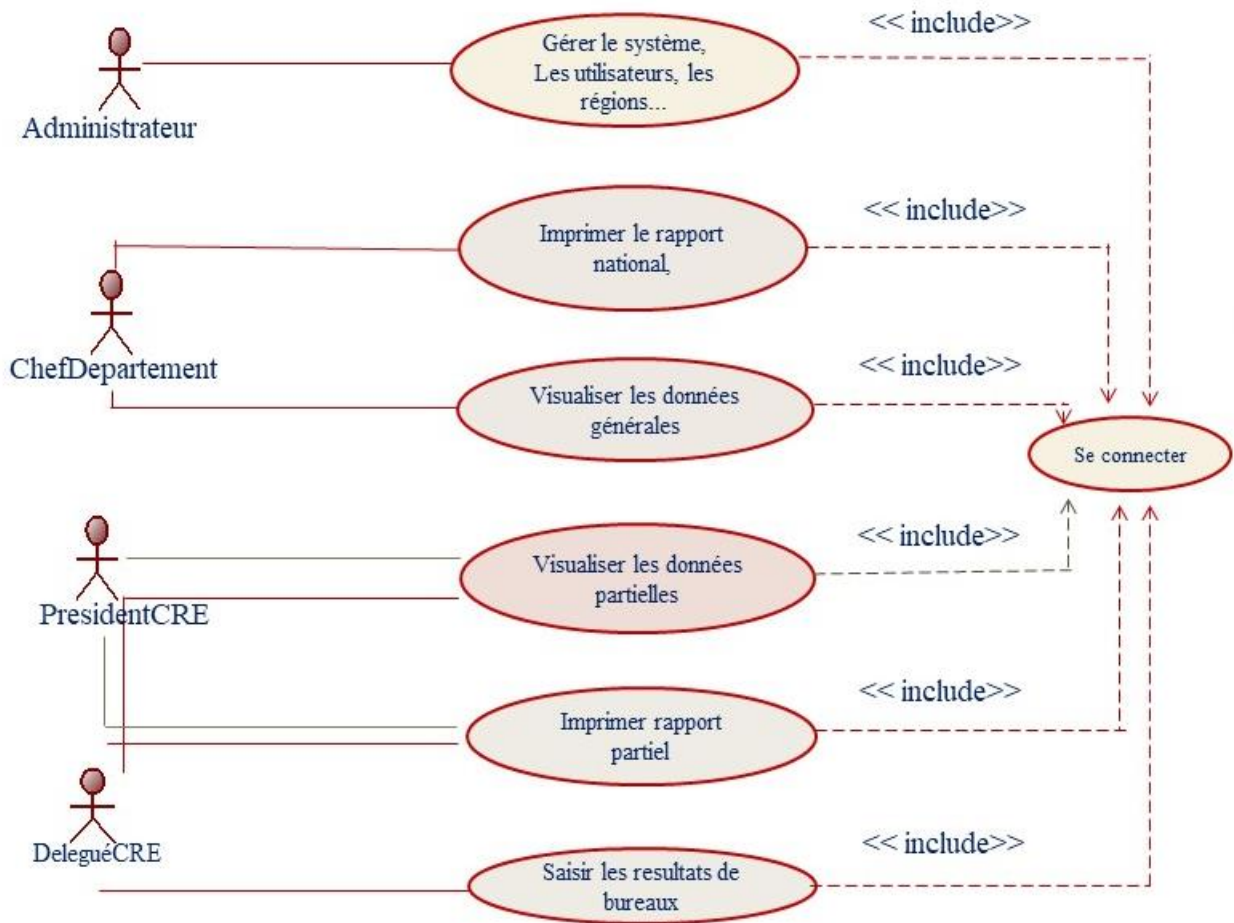


Figure 2 - Diagramme de cas d'utilisation

La figure ci-dessus illustre notre diagramme de cas d'utilisation. Ce dernier est composé de 4 acteurs **Administrateur**, **Chef de Département Statistique et Informatique**, **Président de la Commission Régionale des Élections (CRE)** et **Délégué de la CRE** que nous avons expliqués dans la partie de la description du système.

Pour faire toutes ces actions, ils doivent obligatoirement se connecter au système.

3. Conception du système

a. Diagramme de classe

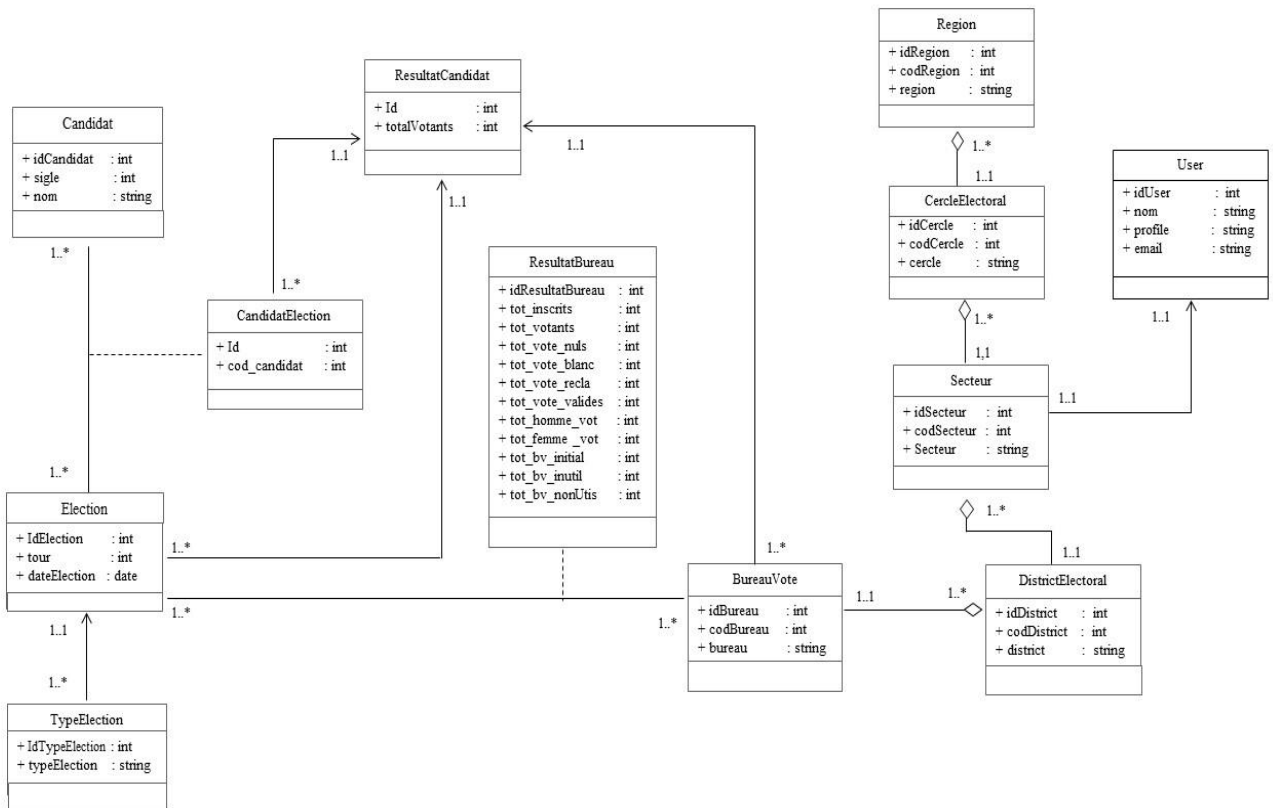


Figure 3 - Diagramme de Classes

Le diagramme de classes est constitué des parties suivantes :

- ✓ La gestion des bureaux de vote : Elle comprend les classes **Region, CercleElectoral, Secteur, DistrictElectoral et BureauVote.**

La relation entre la classe **Region** et la classe **CercleElectoral** signifie qu'une région comporte des cercles électoraux et les cercles électoraux sont dans une région. La relation existante entre les classes **Région et CercleElectoral** est la même qui existe entre les classes **CercleElectoral et Secteur, Secteur et DistrictElectoral, DistrictElectoral et BureauVote.**

- ✓ La gestion des résultats qui est constituée par les classes **CandidatElection, ResultatCandidat et ResultatBureau.**

Dans un bureau de vote, peut se tenir plusieurs élections et pour une **élection**, il peut y avoir plusieurs **Bureaux**. C'est pour cela que nous nous avons la classe d'association **ResultatBureau** pour chaque élection et bureau.

Dans un bureau de vote nous pouvons avoir plusieurs résultats pour les candidats et chaque résultat (voix) d'un candidat d'élection appartient à un **Bureaux**. C'est pour cela qu'il y a une association entre classes **BureauVote** et **RésultatCandidat**.

- ✓ La gestion des élections concernant les classes **Candidat**, **Election** et **TypeElection**.
Une élection a un type et pour chaque type élection on peut organiser plusieurs élections. C'est pour cela qu'on a une association entre **classe TypeElection** et **Election**.
Pour chaque élection organisée, nous avons plusieurs candidats et un candidat peut participer à plusieurs élections. C'est pour cela que nous avons une classe d'association **CandidatElection** entre les classes **Election** et **Candidat**.
- ✓ La gestion des utilisateurs : elle comprend la classe **User**.
La classe **User** est associée à la Classe **Secteur** pour permettre l'utilisateur que va saisir les résultats de bureaux de vote de voir seul les bureaux de secteur qu'il gère.

b. Architecture logicielle

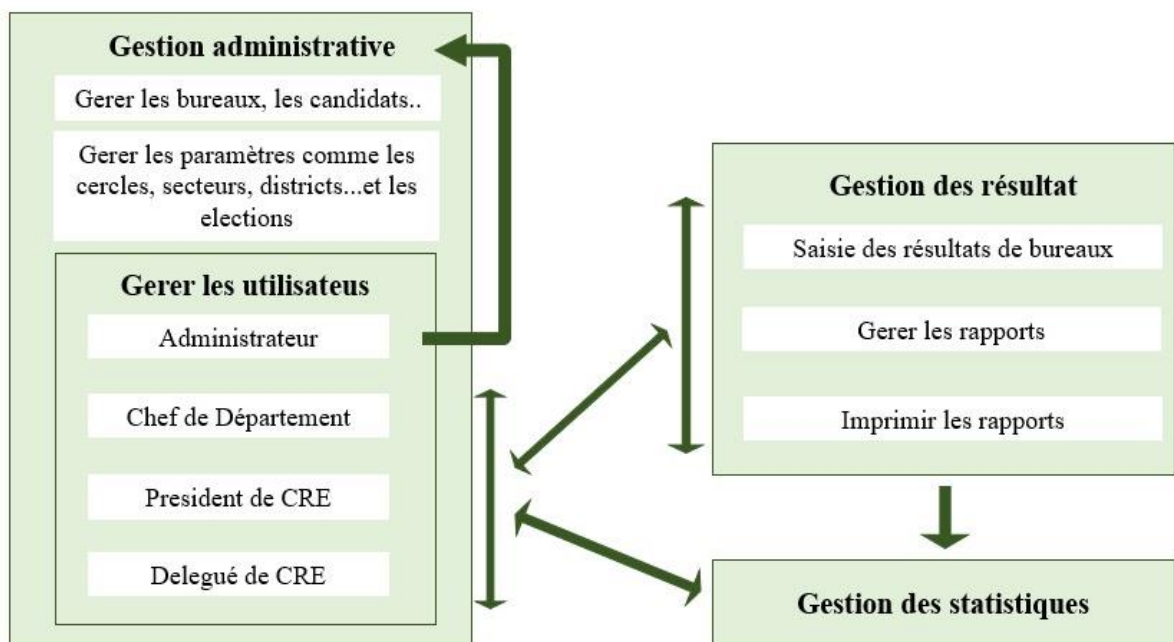


Figure 4- L'Architecture Logicielle

L'architecture logicielle est constituée de trois parties :

- ✓ Une gestion administrative qui regroupe :
 - La gestion des bureaux, les candidats et les élections ;
 - La gestion des paramètres comme la région, les cercles électoraux, secteurs, district électoraux Bureaux de votes, les élections, inscription des candidats...

- La gestion des utilisateurs : ils sont répartis en quatre profils que nous avons précédemment expliqués.
- ✓ Une gestion des résultats qui est constitué de trois tâches pour lesquelles, il y a certaines que certains utilisateurs peuvent faire et autres non.
- ✓ Une gestion des statistiques permettant de visualiser les statistiques de votation.

c. Conclusion

Une étude comparative nous a permis de choisir UML comme outil de modélisation et conception contrairement à Merise. En outre, la description des différents acteurs du système avec leurs rôles et l'élaboration d'un diagramme de cas d'utilisation nous ont permis de mettre en place un diagramme de classes, une architecture logicielle. En fin nous avons mis en place un entrepôt de données, ce qui nous permet de voir d'ores et déjà à quoi ressemble notre application et d'avoir une idée sur la mise en œuvre et les outils technologiques que pouvons utiliser.

Chapitre 3 : ETUDE DES OUTILS TECHNOLOGIQUES

Introduction

Ce chapitre se penche sur les choix des différents outils technologiques que nous utilisons pour la réalisation de notre application web.

En premier lieu, nous faisons une étude comparative de certains langages de programmation d'applications web pour en choisir un. En deuxième lieu, nous faisons la même chose pour choisir un serveur d'application. En troisième lieu, nous procédons à une étude des Frameworks que nous avons utilisé et en fin, nous choisissons un SGBD après une étude comparative de certains d'entre eux.

1. Langages de programmation

a. Java Enterprise Edition (Java EE)



Java EE est une norme proposée par la société Sun, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'application d'entreprise basées sur des composants.

On parle généralement de « plateforme Java EE » pour désigner l'ensemble constitué des services (API pour Application Programming Interface) offerts et de l'infrastructure d'exécution.

Java EE comprend notamment :

- ✓ Les spécifications du serveur d'application, c'est-à-dire de l'environnement d'exécution. Il définit finement les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées. Ces recommandations permettent ainsi à des entreprises tierces de développer des serveurs d'application conformes aux spécifications ainsi définies, sans avoir à redévelopper les principaux services.
- ✓ Des services, au travers d'API, c'est-à-dire des extensions Java indépendantes permettant d'offrir en standard un certain nombre de fonctionnalités. Sun fournit une implémentation minimale de ces API appelée Java EE SDK (Java EE Software Development Kit).

Points forts de Java EE

- ✓ multiples IDE
- ✓ réutilisabilité du code

- ✓ développement site internet
- ✓ intégration des SGBD
- ✓ génération de PDF
- ✓ ...

Points faibles de Java EE

- ✓ lourdeur dans la programmation
- ✓ appropriation du langage et des Frameworks difficile
- ✓ ...

b. PHP

Créé en 1994 comme un simple jeu de scripts CGI (Common Gateway Interface), PHP a gagné en popularité avec PHP3 en 1998, et explose littéralement depuis 2000, avec une version 4 qui en fait l'un des langages



de script les plus utilisés pour les applications côté serveur. Son succès est dû à sa gratuité (et donc sa présence en standard chez de nombreux hébergeurs gratuits, au contraire de l'ASP de Microsoft), sa simplicité d'approche couplée à une certaine puissance, ainsi que son grand jeu de fonctions qui permettent de ne pas réinventer la roue à chaque programme. C'est aujourd'hui le langage le plus présent côté serveur (Web).

Considérée jusqu'à aujourd'hui comme un simple langage de script réservé à la création de site web dynamiques, la plateforme PHP a subi une profonde rénovation pour sa 5^{ème} version. Désormais, PHP 5 offre des fonctionnalités proches de Java EE et .NET.

PHP propose deux syntaxes : l'une procédurale, l'autre orientée objet. Chacune de ces syntaxes permet de mettre en œuvre les mêmes fonctionnalités mais vise des publics différents. La première est destinée aux web masters et aux informaticiens qui travaillent sur l'interface graphique des applications. La seconde syntaxe, orientée objet, est très proche de Java et C# dont elle s'inspire volontairement pour diminuer les coûts de formation des entreprises.

Points fort de PHP

- ✓ simplicité
- ✓ souplesse
- ✓ interopérabilité
- ✓ portabilité

- ✓ pérennité
- ✓ performances et monté en charge
- ✓ support COM, Java et .Net intégré
- ✓ gratuit
- ✓ prise en main facile
- ✓ fonctions de base très nombreuses
- ✓ BDD : utilisation très simple de MySQL, oracle, SQL Server, Sybase, (bibliothèque intégrées), DB2(ODBC).
- ✓ possibilité de créer des PDF
- ✓ utilisation facile d'XML
- ✓ IDE : rapide, mais pas très élaboré
- ✓ création de web services

Points faibles de PHP

- ✓ il manque d'IDE « hauts niveaux »
- ✓ débogage très limité,
- ✓ outils de développement pas à la hauteur du langage
- ✓ mauvaise intégration avec les progiciels (mise à part SAP)
- ✓ pas de container d'objet comme les EJB en java
- ✓ pas d'accès aux bibliothèques du système
- ✓ pas de pools de connexion
- ✓

c. DotNET



DotNET (noté aussi .NET) est un standard proposé par la société Microsoft, pour le développement d'applications d'entreprises. Il est basé sur des composants Microsoft. DotNET constitue ainsi la réponse de Microsoft à la plateforme Java EE de Sun. La plateforme .NET a été élaborée en s'appuyant sur une communauté d'utilisation de spécifications. Ces spécifications ont été ratifiées par un organisme international de standardisation, l'ECMA (European Computer Manufacturers Association), ce qui en fait un standard. Ainsi l'effort de standardisation a permis l'émergence de plateformes portées par des entreprises tierces et disponibles sous un grand nombre de systèmes d'exploitation.

Grace au CLR (Common Language Runtime), la plateforme DotNET est indépendante de tout langage de programmation et supporte un grand nombre de langages de programmation (25) dont le plus utilisé est C# qui s’inspire directement de C et de Java.

Points forts de DotNET

- ✓ programmation très facile
- ✓ simplicité de création de Web Services
- ✓ accès aux bibliothèques système
- ✓ SGBD pour SQL Server
- ✓ support de 25 langages
- ✓

Points faibles de DotNET

- ✓ choix des OS limité
- ✓ faible communauté de développeur
- ✓ intégration à l’existant (interfaçage avec d’autres programmes non Microsoft)
- ✓ connexion aux bases de données non Microsoft coupées toutes les deux minutes en débuge..

2. Choix de langage

Critères	JEE	PHP	DotNET
Mise en place d’une application (en unité de temps)	3U	1U	2U
Interopérabilité (utilisation de fonctionnalités d’application basées sur différentes technologies)	AAA	AAA	A
Taille des projets	Toute taille	Toute taille	Petites et moyennes
Développement site Internet Simple	AA	AAA	AA
Quantité de plateformes et d’OS Compatibles	AAA	AAA	A
Indépendance de la plateforme	AAA	AAA	A
Intégration à l’existant (cohabitation avec d’autres technologies)	AAA	AAA	A

Tableau 2 – Tableau comparatif des plateformes de développement d’applications web

Nous avons choisi, d'après ce tableau comparatif, la plateforme **PHP** du fait de son interopérabilité avec plusieurs autres plateformes, de son indépendance et de sa flexibilité. PHP supporte aussi des projets de toute taille.

3. Serveur d'application

Qu'est-ce qu'un serveur Web ?

Les serveurs de fichiers, les serveurs de bases de données, les serveurs de messagerie et les serveurs web utilisent différents types de logiciels de serveur. Chacune de ces applications peut accéder aux fichiers stockés sur un serveur physique et les utiliser pour différents buts.

Le travail d'un serveur web consiste à héberger des sites web sur internet. Pour atteindre cet objectif, il agit comme un intermédiaire entre le serveur et les machines des clients. Il extrait le contenu du serveur sur chaque requête d'utilisateur et le transmet au web.

Le plus grand défi d'un serveur web est de servir simultanément plusieurs et différents utilisateurs web – chacun demandant des pages différentes. Les serveurs web traitent les fichiers écrits dans différents langages de programmation tels que PHP, Python, Java et autres.

Ils les transforment en fichiers HTML statiques et diffusent ces fichiers dans le navigateur des utilisateurs web. Un serveur web est l'outil responsable de la communication serveur-client. [4]

Qu'est-ce qu'Apache ?



Apache Web Server est **Serveur Apache HTTP** et il est maintenu et développé par Apache Software Foundation.

Il permet aux propriétaires de sites web de servir du contenu sur le web : d'où le nom « serveur web ». C'est l'un des serveurs web les plus anciens et les plus fiables avec une première version sortie il y a plus de 20 ans, en 1995.

Quand quelqu'un souhaite visiter un site web, il saisit un nom de domaine dans la barre d'adresse de son navigateur. Puis le serveur web fournit les fichiers demandés en agissant comme un livreur virtuel.

Comment fonctionne le serveur web Apache ?

Bien que nous appelions Apache un serveur web, ce n'est pas un serveur physique mais plutôt un logiciel qui s'exécute sur un serveur. Son travail consiste à établir une connexion entre un serveur et les navigateurs des visiteurs du site web (Firefox, Google Chrome, Safari, etc.) tout en délivrant des fichiers entre eux (structure client-serveur). Apache est un logiciel multiplateforme, il fonctionne donc à la fois sur les serveurs Unix et Windows.

Lorsqu'un visiteur souhaite charger une page sur votre site web, par exemple, la page d'accueil ou votre « A propos de nous », son navigateur envoie une requête à votre serveur et Apache renvoie une réponse avec tous les fichiers demandés (texte, images, etc.). Le serveur et le client communiquent via le protocole http et Apache est responsable de la communication fluide et sécurisée entre les deux machines.

Apache est hautement personnalisable, car il a une structure basée sur des modules. Les modules permettent aux administrateurs de serveur d'activer ou de désactiver des fonctionnalités supplémentaires. Apache possède des **modules** pour la sécurité, la mise en cache, la réécriture d'URL, l'authentification par mot de passe et encore plus. Vous pouvez également configurer vos propres configurations du serveur via un fichier appelé **.htaccess**, qui est un fichier de configuration Apache.

Apache vs autres serveurs web

Outre Apache, il existe de nombreux autres serveurs web. Chaque application de serveur web a été créée dans un but différent. Bien qu'Apache soit le plus utilisé, il a plusieurs alternatives et concurrents.

Apache vs NGINX



Nginx, prononcé Engine-X, est une application récente de serveur web, lancée en 2004. A ce jour, elle a acquis une certaine popularité auprès des propriétaires de sites web. Nginx a été créé pour résoudre le problème appelé **c10k**, ce qui signifie qu'un serveur web utilisant des fils pour gérer les demandes des utilisateurs ne peut pas gérer plus de 10 000 connexions simultanément.

Apache vs Tomcat



Tomcat est un serveur web également développé par Apache Software Foundation. Son nom officiel est **Apache Tomcat**. C'est un serveur HTTP aussi mais il alimente les applications Java au lieu des sites web

statiques. Tomcat peut exécuter différentes spécifications Java telles que Java Servlet, JavaServer Pages (JSP), Java EL et WebSocket.

1. Tomcat a été créé spécifiquement pour les applications Java, alors qu'Apache est un serveur HTTP à usage général. Vous pouvez utiliser Apache avec différents langages de programmation (PHP, Python, Perl, etc.) l'aide du module Apache approprié (mod_php, mod_python, mod_perl, etc.).
2. Bien que vous puissiez utiliser un serveur Tomcat pour servir également des pages web statiques, il est moins efficace qu'Apache. Par exemple, Tomcat pré-charge la machine virtuelle Java et les autres bibliothèques liées à Java dont vous n'auriez pas besoin sur la plupart des sites web.
3. Tomcat est également moins configurable que les autres serveurs web. Par exemple, pour mettre en marche **WordPress**, le meilleur choix est un serveur HTTP à usage générale tel qu'Apache ou NGINX. [4]

Les avantages et inconvénients d'Apache

Un serveur web Apache peut être un excellent choix pour exécuter un site web sur une plateforme stable et polyvalente. Lorsque les gens recherchent un nouvel hébergement pour leur site web, ils vérifient généralement les limites telles que le stockage, la bande passante, la version PHP utilisée... etc. Alors que nous devrions également faire attention au serveur utilisé pour savoir **quel hébergeur choisir** pour les meilleures performances. Cependant, il faut également faire attention à certains inconvénients d'Apache.

Avantages :

1. Open-source et gratuit même pour un usage commercial.
2. Logiciel fiable et stable.
3. Mise à jour régulière, correctifs de sécurité réguliers.
4. Flexible grâce à sa structure basée sur des modules.
5. Facile à configurer, adapté aux débutants.
6. Plateforme-Cross (fonctionne sur les serveurs Unix et Windows).
7. Fonctionne avec les sites WordPress.
8. Grande communauté et support disponible en cas de problème.

Inconvénients :

1. Problèmes de performances sur les sites web avec un énorme trafic.
2. Trop d'options de configuration peuvent mener à la vulnérabilité de la sécurité.

En définitive, il faut retenir qu'Apache est l'un des serveurs web les plus populaires qui permet de gérer un site web sécurisé sans trop de problèmes. C'est le choix le plus fréquent des solopreneurs et des petites entreprises qui veulent une présence sur le web.

Un site web peut être installé sur un serveur web Apache sans aucune personnalisation. Autrement, Apache fonctionne parfaitement avec de nombreux autres systèmes de gestion de contenu CMS – SGC (**Joomla, Drupal**, etc.), les Framework web (Django, Laravel, etc.) et les langages de programmation. Cela en fait un choix solide pour tous les types de plateformes d'hébergement web. [4]

4. Outils d'optimisation : Les Frameworks

Un framework est un espace de travail modulaire. C'est un ensemble de bibliothèques et de conventions permettant le développement rapide d'applications. Il fournit suffisamment de briques logicielles et impose suffisamment de rigueur pour pouvoir produire une application aboutie et facile à maintenir. Ces composants sont organisés pour être utilisés en interaction les uns avec les autres. Il fournit un ensemble de fonctions facilitant la création de tout ou d'une partie d'un système logiciel, ainsi qu'un guide architectural en partitionnant le domaine visé en modules.

Ainsi, dans le cadre de la réalisation de notre application, nous avons utilisé les Frameworks **Laravel et Bootstrap**. Les parties qui suivent montrent en détails ces différents Frameworks.

a. Laravel



Laravel est un Framework web open-source écrit en PHP respectant le principe **modèle-vue-contrôleur** et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub.

Laravel a été créé par Taylor Otwell en juin 2011, première version 15 juin 2011 et la dernière version 8.27 (10 février 2021).

Laravel s'efforce de fournir une expérience de développement incroyable tout en fournissant des fonctionnalités puissantes telles que l'injection de dépendances approfondie, une couche

d'abstraction de base de données expressive, des files d'attente et des tâches planifiées, des tests unitaires et d'intégration, etc.

Pourquoi Laravel?

Il existe une variété d'outils et de cadres à pour la création d'une application Web. Cependant, nous pensons que Laravel est le meilleur choix pour créer des applications Web modernes et full-stack. [5]

Blade

Blade est le moteur de template utilisé par Laravel. Son but est de permettre d'utiliser du php sur une vue mais d'une manière assez particulière. Pour créer un fichier qui utilise le moteur de template Blade il faut ajouter l'extension ".blade.php". [6]

Blade est le moteur de modélisation simple mais puissant qui est inclus avec Laravel. Contrairement à certains moteurs de création de modèles PHP, Blade ne vous empêche pas d'utiliser du code PHP simple dans vos modèles. En fait, tous les modèles Blade sont compilés dans du code PHP simple et mis en cache jusqu'à ce qu'ils soient modifiés, ce qui signifie que Blade n'ajoute pratiquement aucune surcharge à votre application. Les fichiers de modèle de lame utilisent l'extension de fichier *.blade.php* et sont généralement stockés dans le *resources/views répertoire*. [6]

b. Bootstrap



Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub. [7]

5. Bibliothèques

a. JQuery



JQuery (la bibliothèque JavaScript open-source) vous permet d'exploiter la puissance de JavaScript pour accomplir une myriade de choses géniales sur votre page web.

Avec jQuery, vous pouvez par exemple :

- Ajouter, supprimer ou modifier des éléments HTML au sein de votre page.
- Changer les styles des éléments de la page en modifiant le CSS qui leur est associé.
- Animer des éléments de votre page.
- Envoyer et recevoir des données depuis un serveur grâce à AJAX (Asynchronous JavaScript and XML, c'est-à-dire JavaScript et XML asynchrones) pour ne plus avoir besoin de recharger vos pages après validation d'un formulaire.
- Et bien plus encore !

Par ailleurs, utiliser jQuery vous permet de profiter d'une plus grande **compatibilité avec les différents navigateurs**. Le JavaScript pur est réputé pour avoir des spécificités dans chaque navigateur comme Internet Explorer, Chrome, Safari, etc., qui vont potentiellement exécuter votre code JavaScript différemment les uns des autres. JQuery vous évite cette prise de tête, car il est conçu dans un esprit de compatibilité maximale. [7]

6. Outils techniques : choix du SGBDR

Un système de gestion de base de données (SGBD) est un logiciel système destiné à stocker et à partager des informations dans une base données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Un SGBD permet d'inscrire, de retrouver, de modifier, de trier, de transformer ou d'imprimer les informations de la base de données. Il permet d'effectuer des comptes rendus des informations, éviter des pertes d'informations dues à des pannes, assurer la confidentialité et permettre son utilisation pour d'autres logiciels.

Il existe plusieurs familles de SGBD :

- ✓ SGBD relationnelle :

Un SGBD relationnel est un système dont les utilisateurs voient les données comme un ensemble de tables reliées les unes aux autres par des valeurs qu'elles ont en commun. Les données sont stockées dans des tables composées de lignes et de colonnes. Les tables de données indépendantes peuvent être reliées (ou mis en relation) les unes aux autres, si chacune d'elles dispose d'une colonne de données uniques servant d'identifications (appelées clés) représentant les données qu'elles partagent.

Nous allons maintenant faire une étude comparative de quelques SGBD pour en choisir un qui sera mieux adapté pour stocker les données de notre plateforme.

1- MySQL



MySQL est un système de gestion de bases de données relationnelles rapide, robuste et facile d'utilisation. Il est adapté à la gestion de données dans un environnement réseau, notamment en architecture client/serveur. Il est fourni avec de nombreux outils et est compatible avec nombreux langages de programmation. Il est le plus célèbres SGBDR du monde Open Source (Code source libre), particulièrement grâce à son interopérabilité avec le serveur de pages Web Apache et le langage de pages Web dynamiques PHP. Ce serveur de base de données est interrogeable via SQL (Structure Query Language), le langage standardisé le plus populaire pour interroger les bases de données.

Il est écrit en C/C++ et fonctionne sous de nombreux systèmes d'exploitation (Windows, Linux, Mac OS...).

Il dispose d'un driver :

- ✓ ODBC sous Windows ce que le rend interopérable avec la majorité des langages disponibles sous ce système d'exploitation ;
- ✓ ADO via OLEDB sous Windows pour de meilleures performances qu'ODBC.

2- PostgreSQL

PostgreSQL est un SGBDR robuste et puissant, aux fonctionnalités riches et avancées, capable de manipuler en toute fiabilité de gros volumes de données, même dans des situations critiques, il a une version Open Source et libre et une version commerciale qui s'appelle Red Hat Database. Il fonctionne sous standard SQL sur les principaux systèmes d'exploitation (Linux, Mac, OS X, Windows...). Il utilise un modèle relationnel-objet dans son rapport aux données et permet d'utiliser des fonctions et des règles complexes.



Il gère aussi des langages de procédure en interne, il intègre d'ailleurs un langage natif, appelé PL/pgSQL³, comparable au langage de procédure d'Oracle, PL/SQL⁴

3-Oracle



Oracle Database est un système de gestion de base de données relationnelles, qui, depuis l'introduction du support du modèle objet dans sa version 8, peut être aussi qualifié de système de gestion de base de données relationnel-objet. Fournit par Oracle Corporation, il est leader mondial des bases de données.

Oracle est commercialisé selon trois gammes :

- ✓ Edition Enterprise (Entreprise Edition) ;
- ✓ Edition Standard (Standard Edition) et Edition Standard One (Standard Edition One);
- ✓ Edition Personnelle (Personal Edition) sur plateforme Windows uniquement.

L'édition standard comporte toutes les fonctionnalités de base permettant de mettre en œuvre des applications client-serveur ou Internet/Intranet, pour un groupe de travail ou un département d'entreprise. Cette édition ne permet pas de faire fonctionner les options avancées d'Oracle10g et limité à des serveurs ou des clusters⁵ de serveurs avec une capacité maximale de 4 processeurs. Depuis quelques temps, Oracle commercialise aussi une édition Standard One, fonctionnellement identique à l'édition Standard, mais limité à des serveurs biprocesseurs.

L'édition Enterprise est plus particulièrement destinée aux applications critiques de l'entreprise et propose des fonctionnalités supplémentaires, en standard ou en option, permettant d'améliorer la disponibilité et les capacités de montée en charge des grosses bases de données, et d'en faciliter l'administration et l'optimisation.

³ PL/pgSQL : Procedural Language/PostgreSQL Structured Query Language est un langage procédural géré par PostgreSQL.

⁴ PL/SQL : Procedural Language/PostgreSQL Structured Query Language

⁵ Cluster : c'est une infrastructure virtuelle constituée d'un ensemble de ressources informatiques potentiellement partagées, distribuées, hétérogènes, délocalisées et autonomes.

4- Microsoft SQL Serveur

Microsoft SQL Server est un SGBDR développé et commercialisé par la société par la Microsoft. Il ne fonctionne que sous les OS (Système D'Exploitation) Windows. En fait, il est composé de cinq services principaux :



- ✓ Le moteur relationnel (OLTP) appelé SQL Server ;
- ✓ Le moteur décisionnel (OLAP) appelé SSAS⁶ (SQL Server Analysis Services) incluant un moteur de stockage pour les cubes⁷, des algorithmes de forage (data mining) et différents outils de BI (Business Intelligence) ;
- ✓ Un ETL⁸ (Extract, Transform, Load) appelé SSIS (SQL Server Integration Services) destiné à la mise en place de logiques de flux de données, notamment pour alimenter des entrepôts de données (Datawarehouse) ;
- ✓ Un outil de génération d'état appelé SSRS (SQL Server Reporting Services) permettant de produire des rapports sous différentes formes et exploitant les ressources du moteur décisionnel (base « reportServer... ») à la fois pour y stocker les rapports mais aussi y cacher les données de ces derniers.
- ✓ Un système de planification de travaux et de gestion d'alerte appelé Agent SQL qui utilise lui aussi les services du moteur SQL (base msdb).

6- Tableau comparatif des SGBD

Critères				
SGBD relationnel	Oui	Oui	Oui	Oui
Requete SQL	Oui	Oui	Oui	Oui

⁶ **SSAS** : c'est un système de gestion de données multidimensionnelles.

⁷ **Cube** : c'est une structure de données SQL Server Analysis Services (SSAS) créée à l'aide de bases de données OLAP, qui permet une analyse quasi-instantanée des données. L'intérêt du cube est que les données qu'il contient peuvent être agrégées.

⁸ **ETL** : c'est un processus d'intégration des données qui permet de transférer des données brutes d'un système source, de les préparer pour une utilisation en aval et de les envoyer vers une base de données, un entrepôt de données ou un server cible.

Licence	Open source	Open source	Commerciale	Commerciale
Capacité de stockage	Dépend du support de stockage	Un stockage maximum de 32 Go	Un stockage supérieur à 16 To	524 Po

Tableau 3 - Tableau comparatif des SGBD

D’après le tableau ci-dessus, nous avons choisi le SGBDR MySQL puisqu’il est open source et que sa capacité de stockage est adéquate avec nos besoins. [2]

7. Outils supplémentaires

1. Visual Studio Code



Visual Studio Code est un éditeur de code extensible développé par **Microsoft** pour **Windows, Linux et macOS** .

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la "refactorisation" du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

Le code source de Visual Studio Code provient du projet logiciel libre et open source VSCoDe de Microsoft publié sous la licence MIT permissive, mais les binaires compilés sont des logiciels gratuits pour toute utilisation.

Il est développé par Microsoft, première version disponible le 14 novembre 2015 et la dernière version 1.62.3 (17 novembre 2021).

Fonctionnalités

Visual Studio Code est un éditeur de code source qui peut être utilisé avec une variété de langages de programmation, notamment Java, JavaScript, Go, Node.js, PHP et C++. Il est basé sur le cadre Electron, qui est utilisé pour développer des applications Web Node.js qui s'exécutent sur le moteur de présentation Blink. Visual. [8]

2. WAMP Server



WampServer est une plateforme de développement Web sous Windows. Il permet de développer des sites Web dynamiques à l'aide du serveur **Apache2**, du langage de scripts **PHP5** et d'une base de données **MySQL** version 5. Le fait qu'elle intègre le **SGBD MySQL** nous a permis de l'utiliser pour gérer notre base de données.

Il possède également **PHPMyAdmin** qui est une application web permettant de gérer plus facilement les bases de données.

WampServer s'installe facilement et son utilisation très intuitive permet de le configurer très rapidement (sans toucher aux fichiers de configuration). [9]

Conclusion

Dans ce chapitre, nous avons effectués des études comparatives qui nous ont permis, respectivement, de choisir **PHP** comme langage de programmation de notre application, apache comme server d'application et MySQL comme SGBD. En outre, nous avons fait une étude des Frameworks que nous avons utilisés. Ces derniers ont permis de faciliter la mise en œuvre de notre application.

Chapitre 4 : IMPLEMENTATION ET PRESENTATION DE L'APPLICATION

Introduction

Dans ce dernier chapitre, il s'agit de la mise en œuvre de notre solution.

Ainsi, en premier lieu, nous élaborerons l'architecture applicative de notre application web. En deuxième lieu, nous montrerons la structure de base de données et en fin, nous procéderons aux tests et validation de la solution.

4.1 Architecture applicative 3-tiers de l'application web

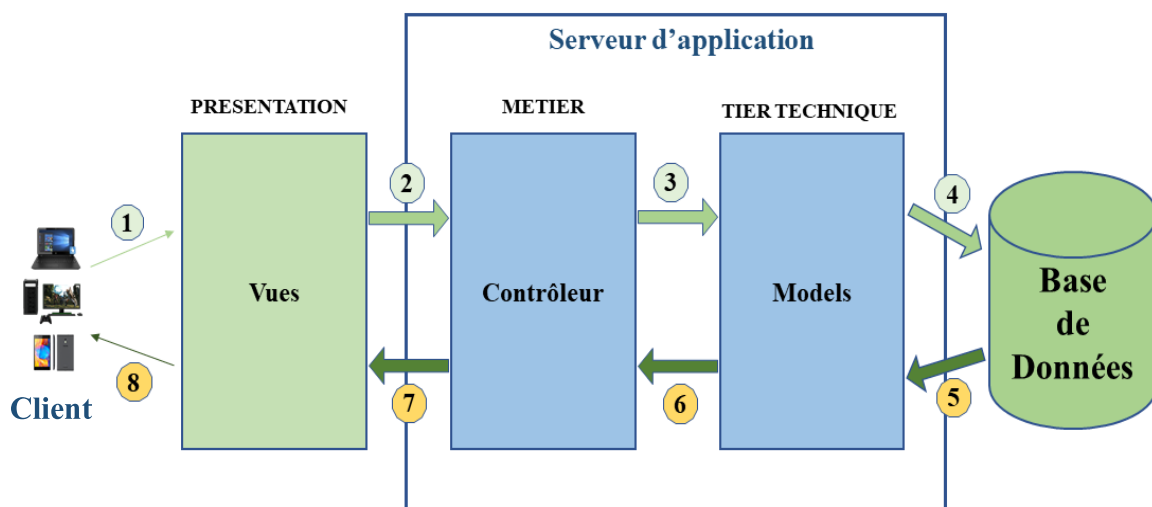


Figure 5 - L'Architecture Applicative

L'architecture applicative sera conforme au modèle MVC (Model, View, Controller en Anglais). Il décrit une manière d'architecturer une application informatique en la décomposant en trois sous-parties :

- ✓ La partie **Modèle** : encapsule la logique métier ainsi que l'accès aux données. Il peut s'agir d'un ensemble de fonctions (Modèle procédural) ou de classes (Modèle orienté objet).
- ✓ La partie **Vue** : elle s'occupe des interactions avec l'utilisateur ; présentation, saisie et validation des données.
- ✓ La partie **Contrôleur** : elle gère les traitements, donc la dynamique de l'application.

L'application, une fois réalisée, sera déployée dans un serveur d'application afin qu'elle puisse être accédée par les clients via des navigateurs Internet comme Chrome, Firefox, Torch, Microsoft Edge...

4.2 Structure de la base de données

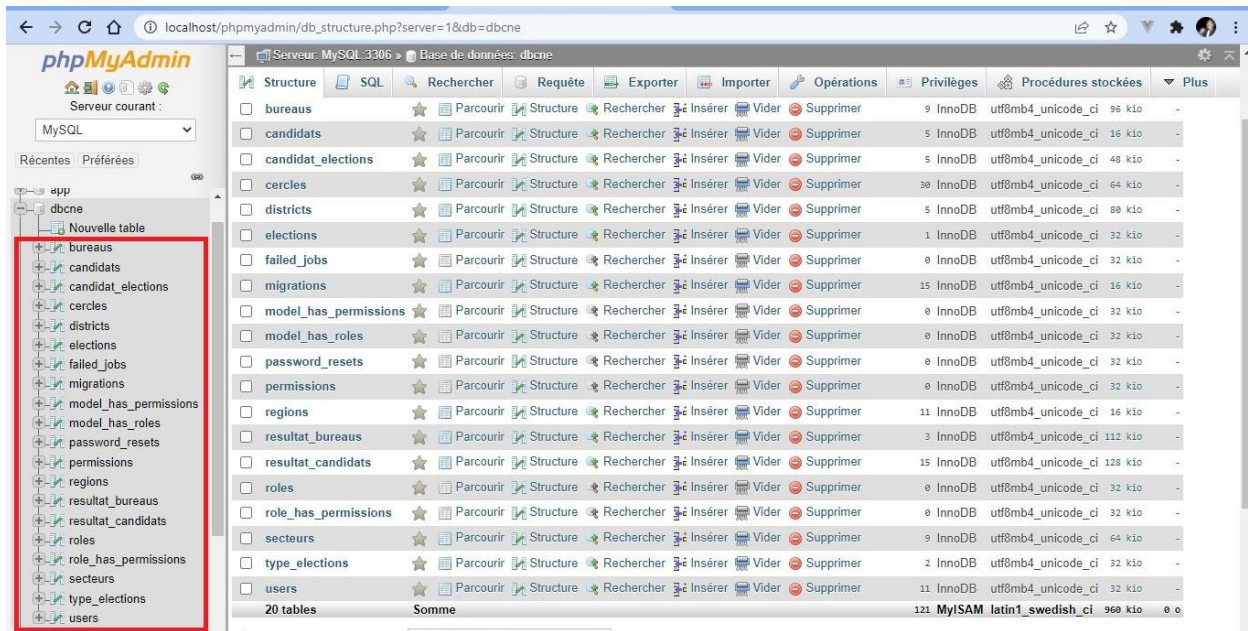


Figure 6 - Structure de la Base de Données

Notre base de données s'appelle **dbcne** et est constituée de 20 tables dont 5 sont pour la gestion des rôles et permissions des utilisateurs et les restes provenant des migrations voir **Annexe 4**.

4.3 Structure du projet

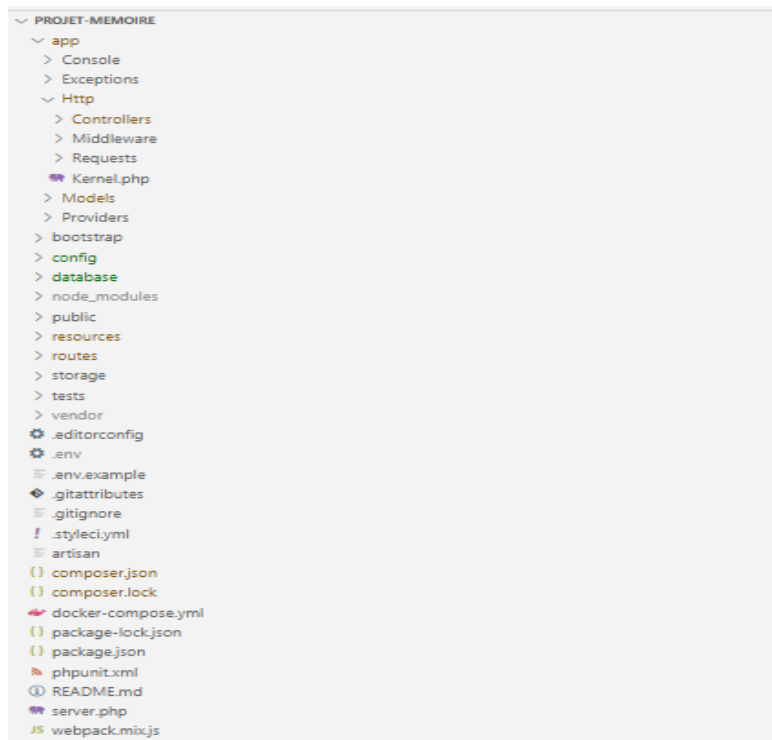


Figure 7-Structure de projet

Le projet s'appelle **PROJET-MEMOIRE** et est structuré en plusieurs répertoires parmi lesquels il y a :

- Le répertoire **app** contenant plusieurs répertoires parmi lesquels il y a répertoires **Http** et **Models**.

Le répertoire **Http** contient l'ensemble des répertoires principal de l'application. Ces répertoires sont répartis dans trois sous répertoires à savoir :

- **Controllers** qui contient les contrôleurs ; c'est-à-dire es classes **contrôleurs** qui jouent le rôle d'intermédiaire entre classes d'accès aux données et les vues (présentation de données).
- **Middleware** contenant les classes où on gère la sécurité applicative donc authentification.
- **Requests** qui contient l'ensemble des classes qui vérifient les données saisies par l'utilisateur au formulaire et les valider.

Le répertoire **Models** contenant les classes qui définissent les différentes méthodes d'accès aux données de la base.

- Le répertoire **database**. Il contient un sous répertoire appelé **migrations** contenant l'ensemble des classes permettant la génération des entités ; c'est-à-dire correspondant aux différentes tables de la base de données.
- Le répertoire **public** contenant plusieurs sous répertoires parmi lesquels **img** qui contient les images, le fichier **CSS** utilisées sur le projet et autres.
- Le répertoire **ressources** contenant aussi plusieurs sous répertoires parmi lesquels on répertoire **views** qui contient le Template que nous avons utilisé et les différentes pages (ou vues) de présentation.
- Le répertoire **routes** contenant le fichier **web** où est défini les routes c'est-à-dire le chemin pour les différentes méthodes permettant l'accès aux données et autres pages de l'application.

Le fichier **.env** qui se trouve dans dossier de projet gère la connexion entre l'application web et la base de données et autres paramètres de la connexion.

4.4 Présentation de l'application

Vu que nous avons 04 profils d'utilisateur, nous allons montrer quelques actions que chacun d'eux peut faire dans notre application web.

Profil Administrateur

L'administrateur est celui qui est habilité à ajouter, modifier et supprimer des utilisateurs, des régions, des cercles, des districts, des bureaux et définir le paramétrage nécessaire de l'application.

Nous allons nous connecter avec le compte administrateur avec comme email : **admin@cne.org** et mot de passe **123456789** pour montre quelques paramètres de l'application.

Voici la page de connexion pour tous les utilisateurs de notre application :



Figure 8-Page de connexion utilisateur

Après connexion, l'administrateur est redirigé vers sa page d'accueil qui est la suivante :



Figure 9-Page accueil administrateur

Maintenant l'administrateur peut voir le statistique d'élection courante en cliquant sur « **Dashboard** », il peut traiter les réclamations en cliquant sur le bouton « **Réclamation** » et pour la gestion des régions, cercles, secteur, districts et bureau il clique sur le bouton « **Localité** » comme le montre la figure suivante :

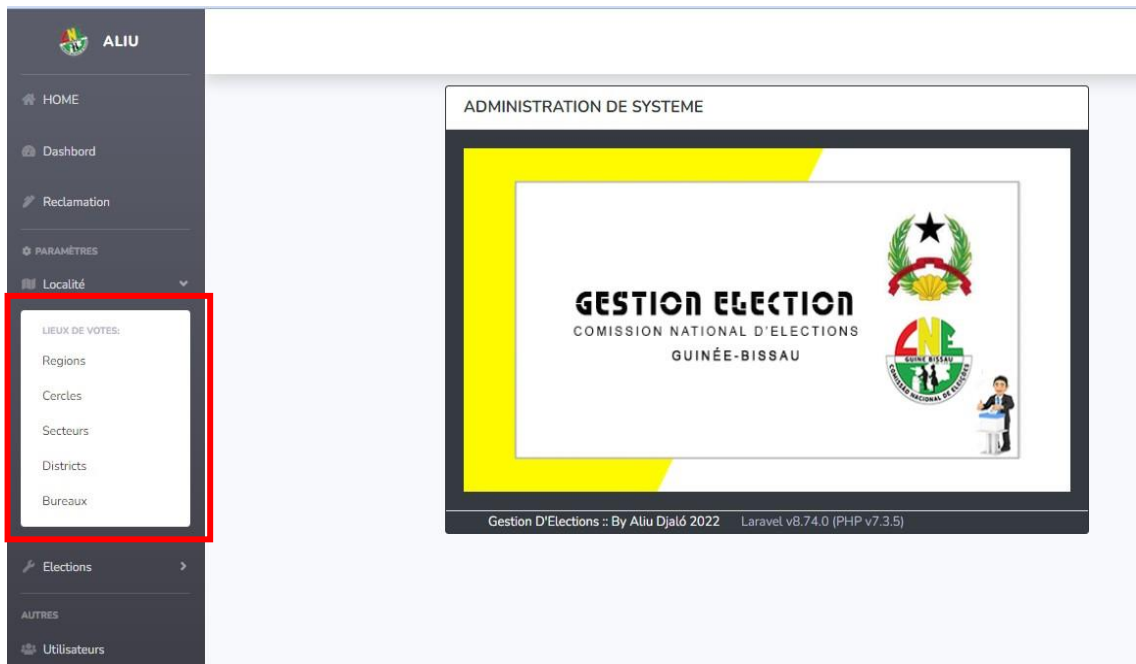


Figure 10-Menu localités

Nous allons prendre comme exemple la gestion des bureaux de vote en cliquant sur bouton « **Bureaux** » de menu comme le montre la figure suivante :

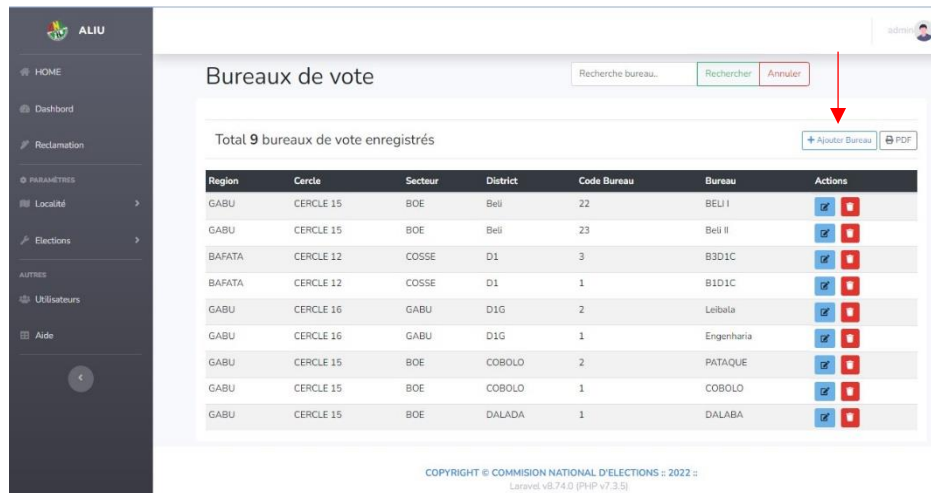


Figure 11-Bureaux de votes

L'administrateur peut ajouter un bureau de vote en cliquant sur « **Ajouter Bureau** », comme le montre la figure suivante :

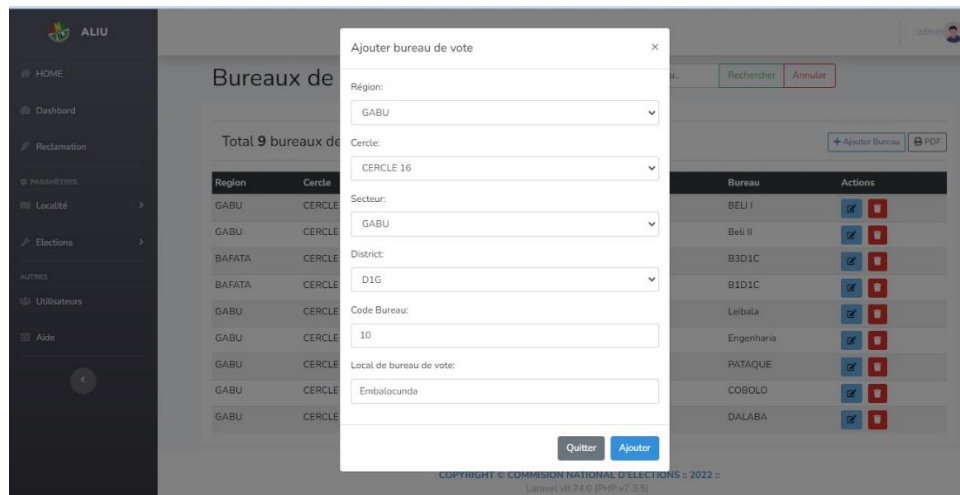


Figure 12- Ajout de bureau de vote

Après renseignement des informations nécessaires, il clique sur « **Ajouter** » puis le bureau de vote est ajouté comme le montre la figure suivante :

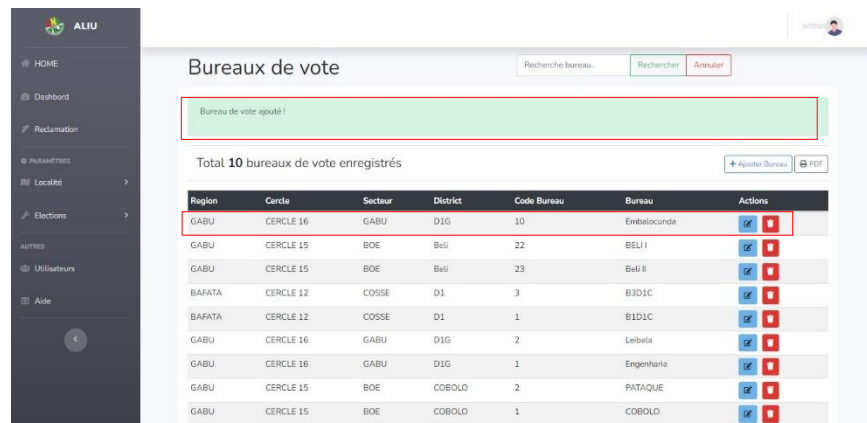


Figure 13-Ajout de bureau avec succès

Pour modifier les informations d'un bureau de vote il clique sur bouton « **stylo** » de menu « **Actions** ».

Après avoir modifié les informations nécessaires, il peut valider la modification en cliquant sur le bouton « **Mettre à jour** ».

Pour supprimer les informations d'un bureau de vote il clique sur bouton [Delete] dans la menu « **Actions** », il va afficher une boîte de confirmation. Il peut valider la suppression de registre bureau en cliquant sur « **OK** ».

L'administrateur peut gérer les élections c'est-à-dire programmer une élection et gérer les candidats... Pour ce faire il clique sur le bouton « **Elections** » comme le montre la figure suivante :



Figure 14- Menu gestion élection

Nous allons prendre comme exemple la gestion des élections en cliquant sur bouton « **Elections** » de menu comme le montre la figure suivante :

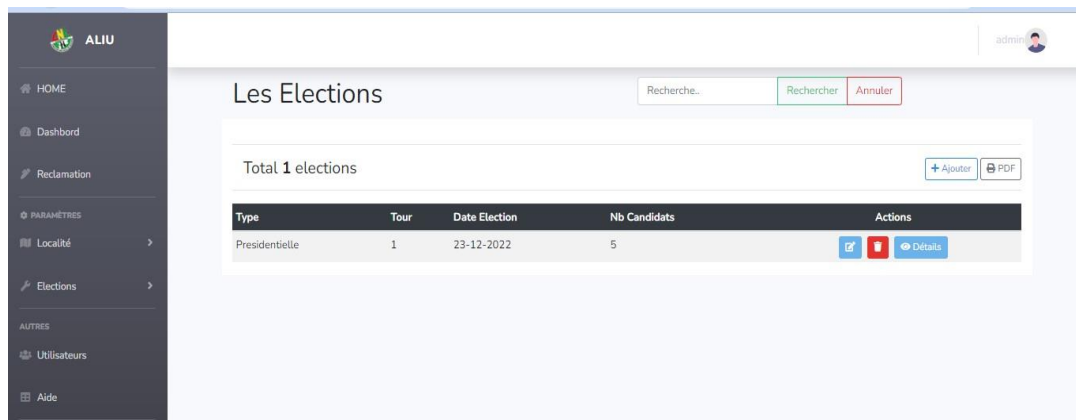


Figure 15- Elections

L'administrateur peut ajouter un bureau de vote en cliquant sur « **Ajouter** », comme le montre la figure suivante :

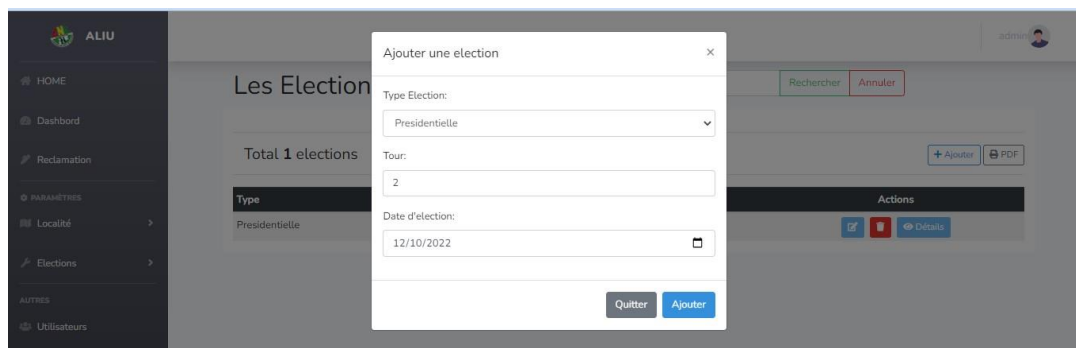


Figure 16-Ajout d'élection

Après renseignement des informations nécessaires, il clique sur « **Ajouter** » puis élection est ajouté avec 0 nombre de candidats comme le montre la figure suivante :

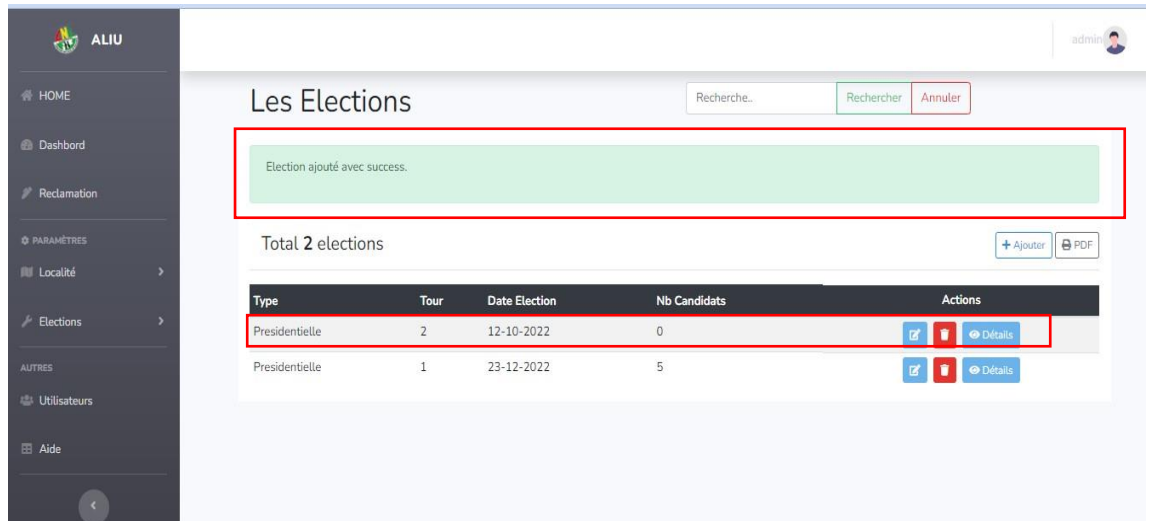


Figure 17- Election ajouté avec succès

Pour modifier les informations d'une élection on clique sur bouton [Edit] dans le menu « **Actions** ». Nous allons modifier la date d'élection.



Figure 18- Modification des informations d'élection

Après avoir modifié les informations nécessaires, il peut valider la modification en cliquant sur le bouton « **Mettre à jour** ».

Pour supprimer les informations d'un bureau de vote il clique sur bouton « **Corbeille rouge** » dans le menu « **Actions** », il va afficher une boîte de confirmation après avoir confirmé avec bouton « **OK** » de boîte, l'élection est supprimée comme montre la figure suivante :

Après chaque élection, l'administrateur pourra afficher les détails en cliquant sur le bouton « **détails** ».

L'administrateur peut faire la gestion des utilisateurs à partir de sa page d'accueil en cliquant sur utilisateurs puis il est redirigé vers la page suivante :

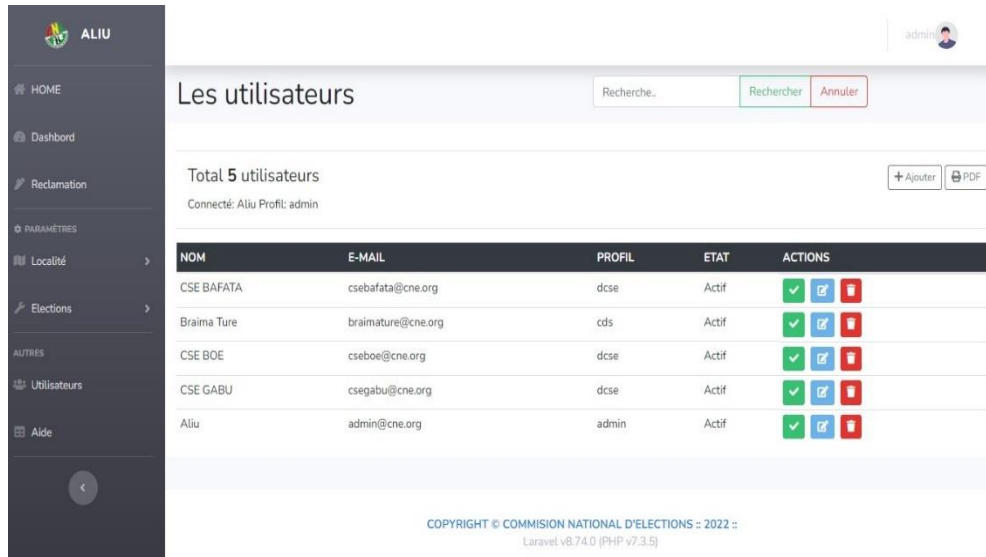


Figure 19- Gestion des utilisateurs

A partir de cette page, l'administrateur peut faire la gestion des utilisateurs. Par exemple, créons un nouvel utilisateur en cliquant sur le bouton « **+Ajouter** ».

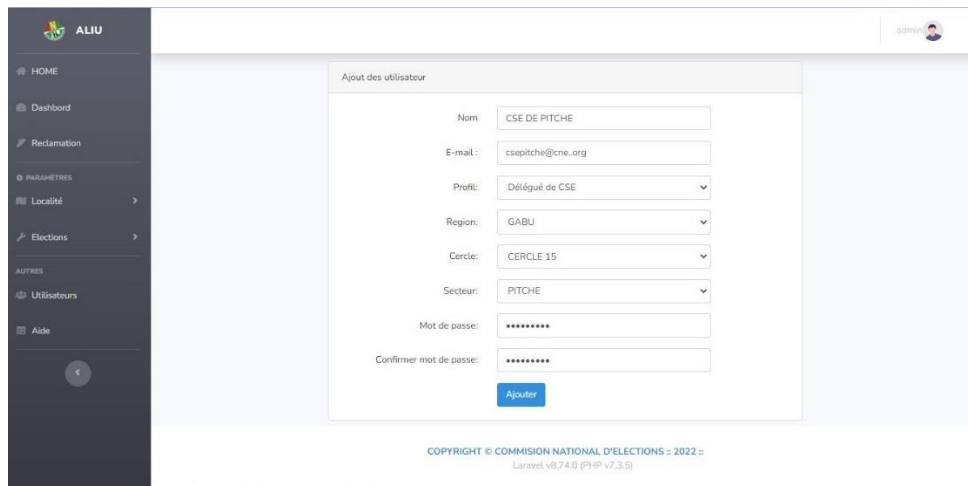


Figure 20-Ajout d'un nouvel utilisateur

Après renseignement des informations d'utilisateur, on clique sur bouton « **Ajouter** ». L'utilisateur est ajouté avec succès.

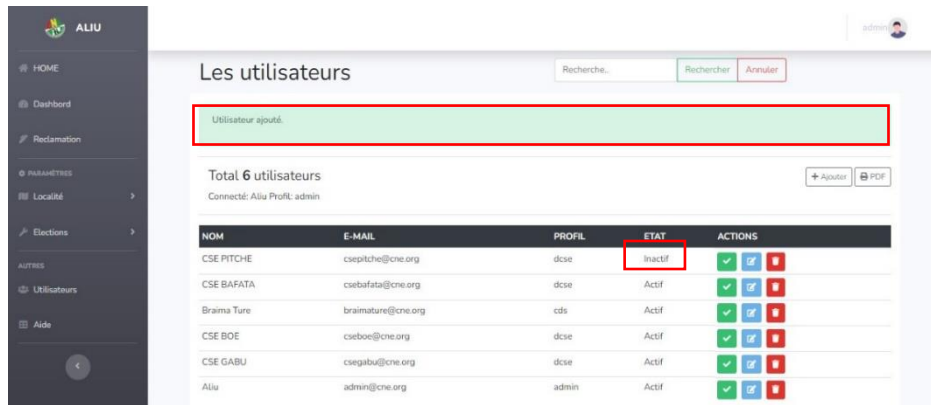


Figure 21- Ajout réussi d'un nouvel utilisateur

L'utilisateur créé, est par défaut **Inactif**, l'administrateur peut l'activer en cliquant sur le bouton « **de couleur verte** » trouvant sur menu « **Actions** ».

L'administrateur peut aussi modifier ou supprimer un utilisateur. Par exemple il peut modifier l'utilisateur précédemment créé en cliquant sur l'icône de couleur « **bleu avec stylo** » ou bien le supprimer en cliquant sur l'icône de « **bouton rouge** ».

Profil Chef Département de Statistique et Informatique

On a dit que le chef de département peut voir le statistique national et imprimer les rapports de résultats électoraux.

Nous utilisons le compte ayant comme email **aliucds@cne.org** et mot de passe **123456789** pour voir ce qu'il peut faire dans notre application.

Une fois qu'il se connecte, voici sa page d'accueil :



Figure 22- Page accueil chef département

Le chef de département de statistique et informatique peut voir le statistique national d'élection en cliquant sur « **Dashboard** » dans la barre de menu.

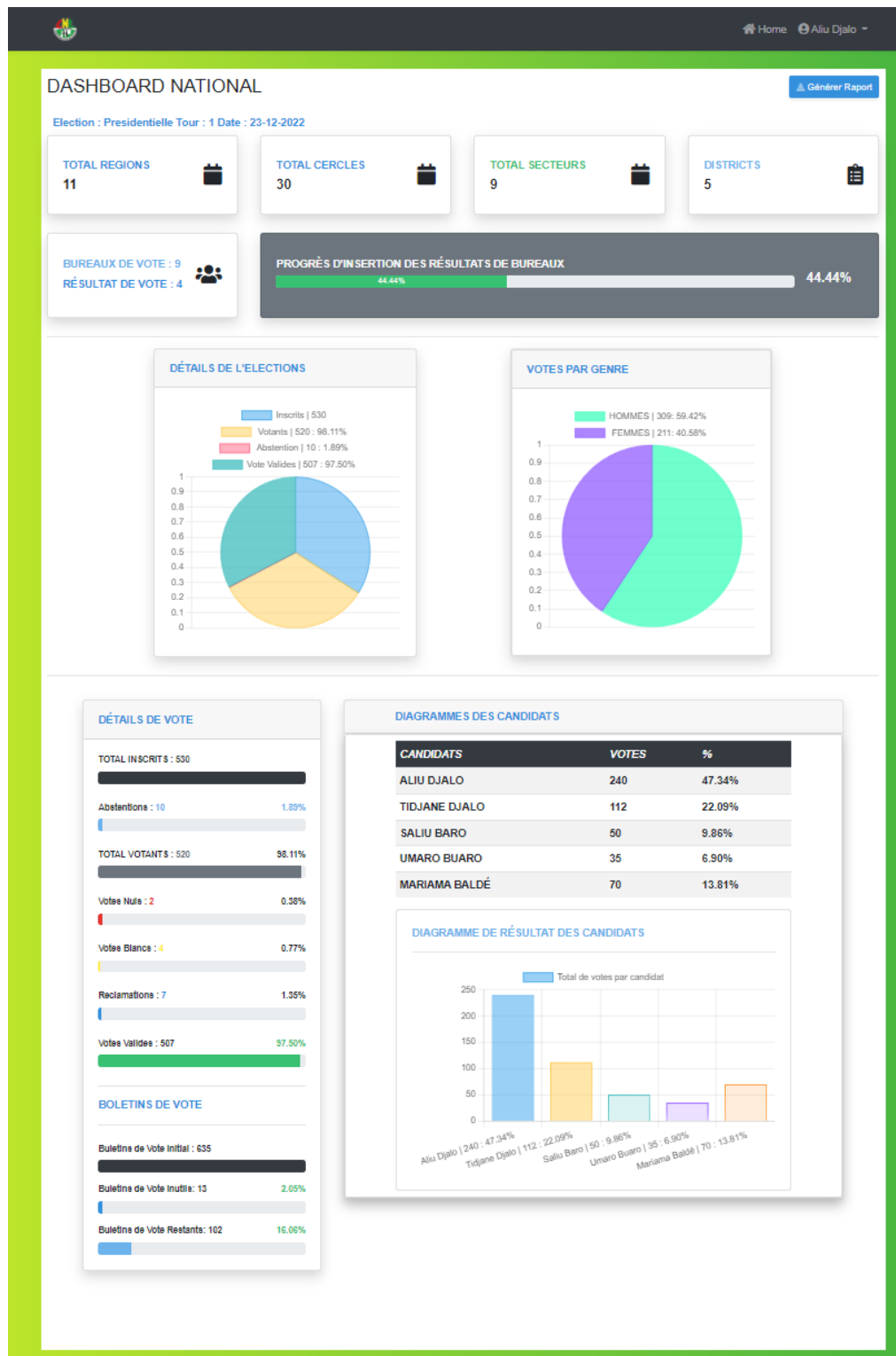


Figure 23-Dashboard National

Chapitre 4 - Implémentation et Présentation de l'Application

Le chef de département de statistique et informatique peut imprimer les rapports nationaux, Régionaux, par cercle électoral, par districts électoral, ... des élections en cliquant sur « **Rapports** » dans la barre de menu.

Voici la Page d'accueil pour gérer les rapports des élections :

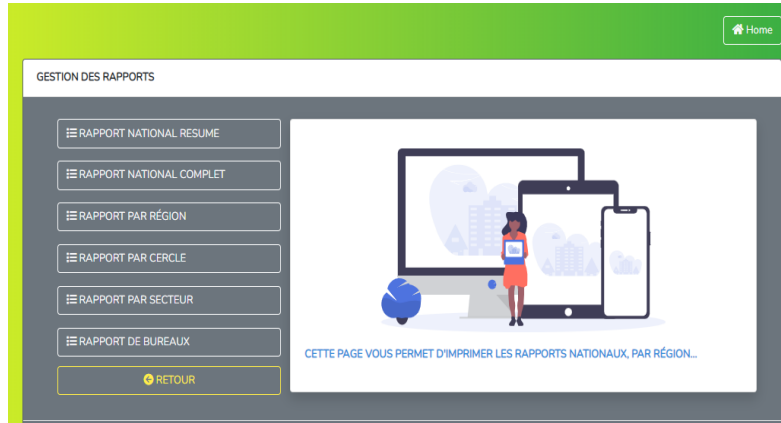


Figure 24-Page d'accueil pour gestion des rapports d'élection

Le chef de département est chargé aussi d'imprimer les rapports nationaux de résultats électoraux. Pour cela, il clique sur le bouton « **RAPPORT NATIONAL RESUME** » ou « **RAPPORT NATIONAL COMPLET** », comme le montre la figure suivante :

REPUBLICQUE DE LA GUINEE-BISSAU ASSEMBLEE NATIONALE POPULAIRE COMMISSION NATIONAL D'ELECTIONS Résultat National d'Election	
Election: Présidentielle Tour: 1 Date : 23-12-2022 Imprimée le: 05-03-2022	
TOTAL REGIONS	11
TOTAL CERCLES	30
TOTAL SECTEURS	9
TOTAL DISTRICTS	5
TOTAL BUREAUX	9
NUMERO D'INSCRITS	530
NUMERO DE VOTANTS	520
NUMERO DE VOTES BLANCS	4
NUMERO DE VOTES NULS	2
NUMERO DE RECLAMATIONS	7
NUMERO DE VOTES VALIDES	507
NUMERO DE L'ABSTENTION	10
HOMMES	309
FEMMES	211
CANDIDATS :	
N° PRENOM NOM	VOTES
1 ALIU DJALO	240
2 TIDJANE DJALO	112
3 SALIU BARO	50
4 UMARO BUARO	35
5 MARIAMA BALDÉ	70
BULLETINS DE VOTES INITIAUX	635
BULLETINS DE VOTES INUTILES	13
BULLETINS DE VOTES RESTANTS	102

Figure 25-Rapport national de résultats électoraux

Chapitre 4 - Implémentation et Présentation de l'Application

Le chef de département peut aussi imprimer les rapports par région en cliquant sur le bouton « **RAPPORT PAR REGION** » qui va afficher le menu de gestion des rapports régionaux comme le montre la figure suivante :











Code	Region	Cercle	Secteurs	Districts	Bureaux	Actions
10	BISSAU	6	0	0	0	 
9	DIASPORA-EUROPA	1	0	0	0	 
9	DIASPORA-AFRICA	1	0	0	0	 
8	CACHEU	3	0	0	0	 
7	GABU	4	5	4	7	 

Figure 26-Page menu rapports régionaux

Pour imprimer le rapport de la région souhaitée, il clique sur l'icône d'imprimante ou sur l'icône d'yeux pour l'afficher. Nous allons imprimer les résultats de la région de **GABU** comme le montre la figure suivante :

REGION	GABU
TOTAL CERCELES	4
TOTAL SECTEURS	5
TOTAL DISTRICTS	4
TOTAL BUREAUX	7

NUMERO D'INSCRITS	380
NUMERO DE VOTANTS	375
NUMERO DE VOTES BLANCS	4
NUMERO DE VOTES NULS	2
NUMERO DE RECLAMATIONS	2
NUMERO DE VOTES VALIDES	367
NUMERO DE L'ABSTENTION	5

HOMMES	220
FEMMES	155

CANDIDATS :		
N°	PRENOM NOM	VOTES
1	ALIU DJALO	200
2	TIDJANE DJALO	92
3	SALIU BARO	35
4	UMARO BUARO	20
5	MARIAMA BALDÉ	20

BULLETINS DE VOTES INITIAUX	455
BULLETINS DE VOTES INUTILS	9
BULLETINS DE VOTES RESTANTS	71

Figure 27-Rapport région de Gabú

Le chef de département peut faire la même procédure au menu principal de gestion des rapports pour l'impression des rapports des résultats électoraux par cercle électoral et par secteur.

Profil de Président de la Commission Régional des Élections

Les utilisateurs qui ont ce profil sont chargés gérer les rapports régionaux. Utilisons le compte ayant come email **cregabu@cne.org** et mot de passe **123456789** pour voir ce qu'il peut faire dans notre application. Une fois qu'il se connecte, voici sa page d'accueil :

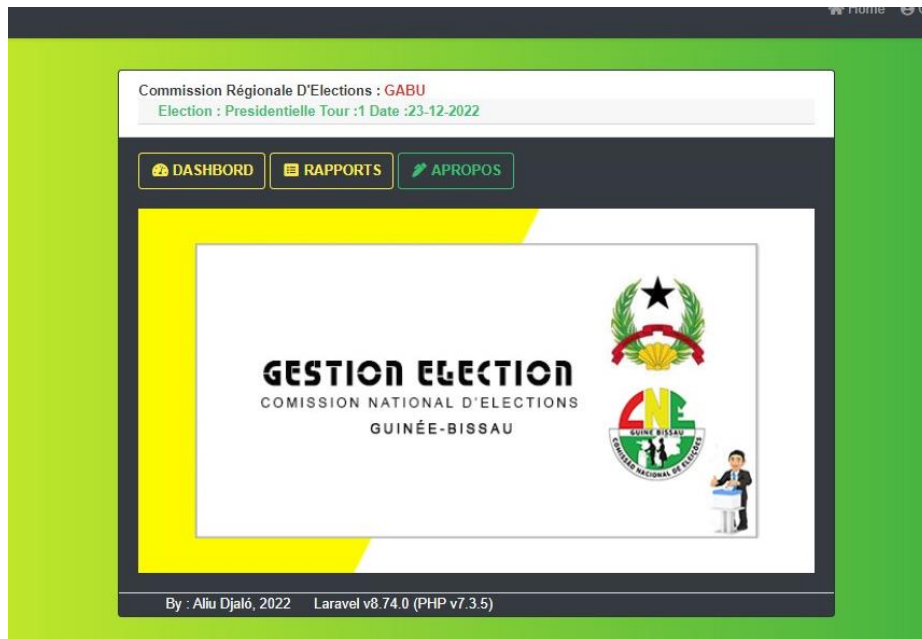


Figure 28-Page Accueil Président de la Commission Régional D'Elect ion

Le président peut voir le statistique de la région où il est affecté. Pour cela il clique sur bouton « **DASHBOARD** » et sera redirigé vers ladite page.

Le président de la CRE peut imprimer les rapports régionaux de résultats électoraux de la région où il est affecté. Pour cela il va cliquer sur le bouton « **RAPPORTS** », après il va voir la page permettant la gestion des rapports régionaux.

Voici la page d'accueil pour la gestion des rapports régionaux :

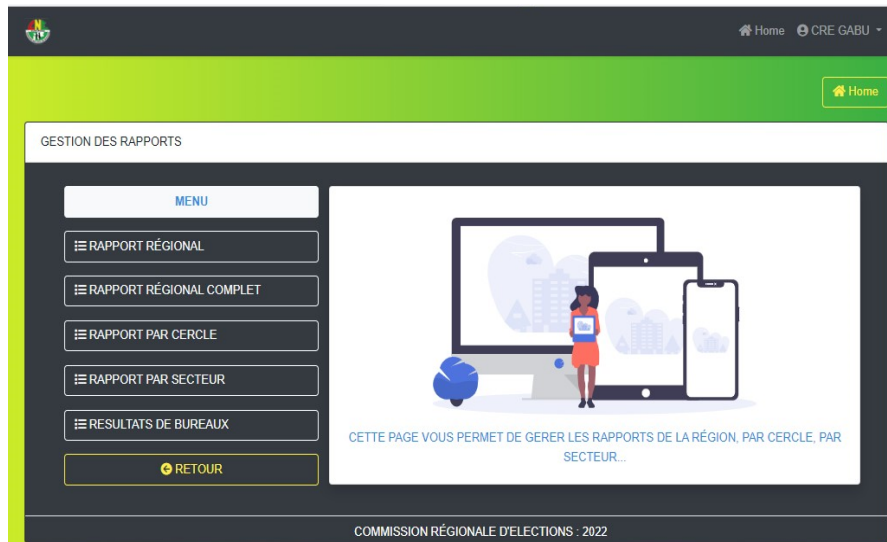


Figure 29-Page Accueil Gestion de Rapports Régionaux

Il peut imprimer le rapport régional complète ou résumé en cliquant sur « **Rapport Régional** » ou « **Rapport Régional Complet** dans la barre de menu.

Ce même profil peut imprimer les rapports des résultats électoraux par cercle électoral de la région qu'il gère en cliquant sur « **RAPPORT PAR CERCLE** » dans la barre de menu.

Voici le page d'accueil pour les cercles :









Code	Cercles	Secteurs	Districts	Bureaux	Action
15	CERCLE 15	2	3	5	 
16	CERCLE 16	1	1	2	 
17	CERCLE 17	1	0	0	 
18	CERCLE 18	1	0	0	 

Figure 30-Page gestion rapport cercle

Il peut imprimer le rapport de cercle désiré en cliquant sur l'icône d'imprimante ou le visualiser en cliquant sur l'icône d'yeux. Voici l'impression de rapport de résultats électoraux **Cercle 15** :

REPUBLIQUE DE LA GUINEE-BISSAU
ASSEMBLEE NATIONALE POPULAIRE
COMMISSION NATIONAL D'ELECTIONS
Résultat partiel de votation

Election: **Présidentielle** Tour: 1 Date : 23-12-2022 Imprimée le: 03-03-2022

REGION	GABU
CERCLE	15
TOTAL SECTEURS	2
TOTAL DISTRICTS	3
TOTAL BUREAUX DE VOTE	5
NUMERO D'INSCRITS	150
NUMERO DE VOTANTS	145
NUMERO DE VOTES BLANCS	1
NUMERO DE VOTES NULS	0
NUMERO DE RECLAMATIONS	0
NUMERO DE VOTES VALIDES	144
NUMERO DE L'ABSTENTION	5

HOMMES	75
FEMMES	70

CANDIDATS :

N° PRENOM NOM	VOTES
1 ALIU DJALO	50
2 TIDJANE DJALO	44
3 SALIU BARO	10
4 UMARO BUARO	20
5 MARIAMA BALDÉ	20

BULLETINS DE VOTES INITIALS	175
BULLETINS DE VOTES INUTILS	0
BULLETINS DE VOTES RESTANTS	30

Figure 31-Rapport de Résultats Electoraux de Cercle 15

Ce même profil peut imprimer les rapports des résultats électoraux par **secteur** de la région qu'il gère en cliquant sur « **Rapport par Secteur** » dans la barre de menu.

Voici le page d'accueil pour les rapports par secteur :

Region : GABU

Election : Présidentielle Tour: 1 Date : 23-12-2022 GESTION DES RAPPORTS PAR SECTEUR

Code	Secteur	Cercle	Districts	Bureaux	Action
1	BOE	CERCLE 15	3	5	
2	PITCHÉ	CERCLE 15	0	0	
3	GABU	CERCLE 16	1	2	
4	PIRADA	CERCLE 17	0	0	
5	SONACO	CERCLE 18	0	0	

Figure 32-Page Accueil gestion rapport Secteurs à la CRE

Il peut imprimer le rapport de **secteur** désiré en cliquant sur l'icône d'imprimante ou le visualiser en cliquant sur l'icône d'yeux.

Ce profil peut aussi imprimer le rapport de résultats de bureau de la région

Profil de Délégué

C'est ce profil qui est chargé de saisir les résultats électoraux et gérer les rapports sectoriels des résultats. Utilisons le compte ayant come email **cseboe@cne.org** et mot de passe **123456789** pour voir ce qu'il peut faire dans notre application.

Une fois qu'il se connecte, voici sa page d'accueil :

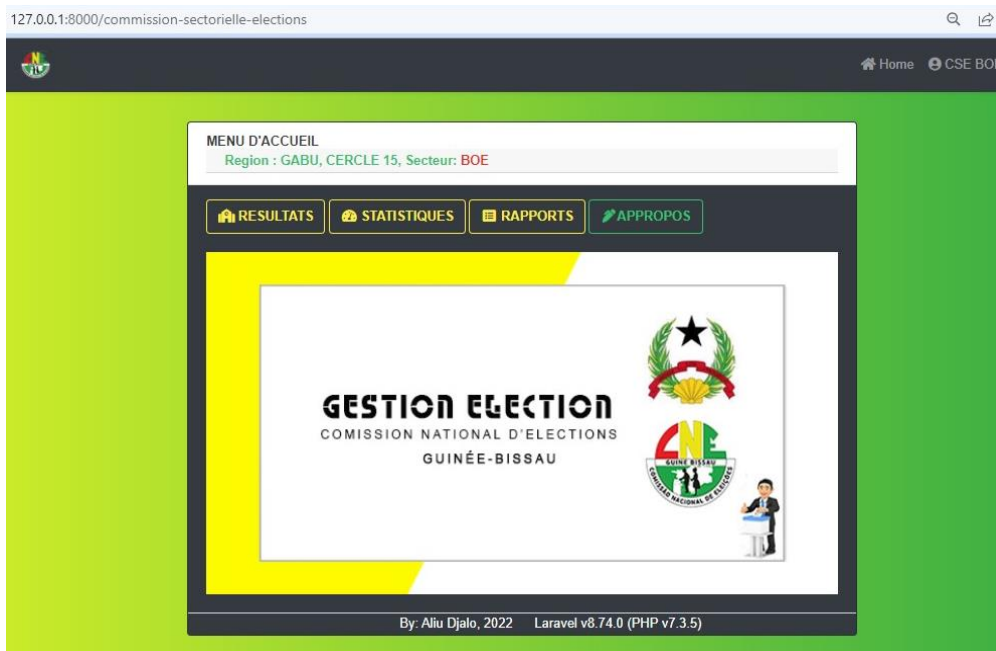


Figure 33- Page accueil Délégué

Pour saisir les résultats de bureaux de vote, il clique sur bouton « **RESULTAT** », il sera renvoyé au module de saisie sectorielle des résultats électoraux comme le montre la figure suivante :

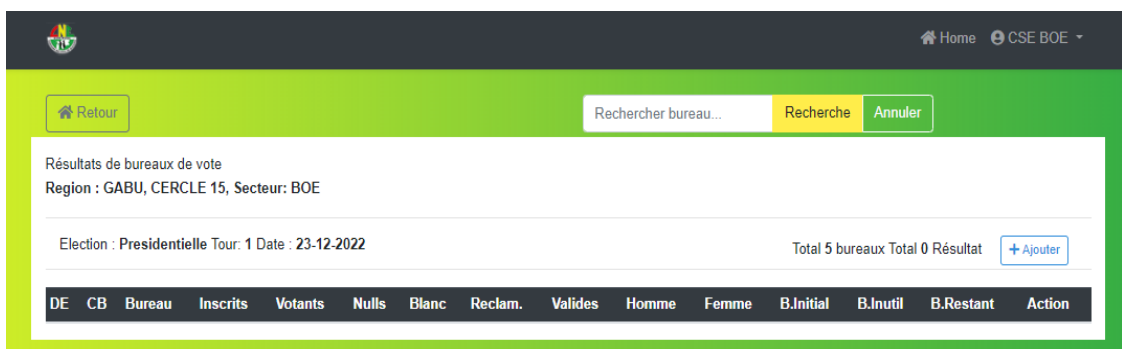


Figure 34-Visualisation sectorielle de résultats

Pour saisir le résultat, il clique sur bouton « **+ajouter** », après il sera renvoyé vers la page de saisie comme le montre la figure suivante :

I:8000/resultats-bureau/create

Ajouter les Résultats Retour

ID : 2 Election : Présidentielle Tour: 1 Date : 23-12-2022

LOCALITE

REGION : GABU

CERCLE : CERCLE 15

SECTEUR : BOE

District Electoral : DALADA

Bureau de Vote : DALABA

DETAILS DE VOTE

Total Electeurs Inscrits : 150

Total Electeurs Votants : 145

Total Votes Blancs : 1

Total Votes Nuls : 0

Total Votes Reclamé : 0

Total Votes Valides : 144

VOTE PAR GENRE

Total Hommes Votants : 75

Total Femmes Votantes : 70

CANDIDATS:

1 : Aliu Djaló : 50

2 : Tidjane Djaló : 44

3 : Saïlu Baro : 10

4 : Umaraou Buaro : 20

5 : Mariama Baldé : 20

BULLETINS DE VOTE

Total Bulletins de Vote Initial : 175

Total Bulletins de Vote Inutiles : 0

Total Bulletins de Vote Restants : 30

Valider Annuler

Figure 35-Page saisie de résultat de vote

Après avoir saisi les informations, l'utilisateur les valide afin de les enregistrer en cliquant sur le bouton « **Valider** » ou Annuler l'opération en cliquant sur le bouton « **Annuler** ».

Après la saisie de résultat il sera redirigé vers la page de visualisation des résultats avec un message "Données enregistrées avec succès" comme le montre la figure suivante :

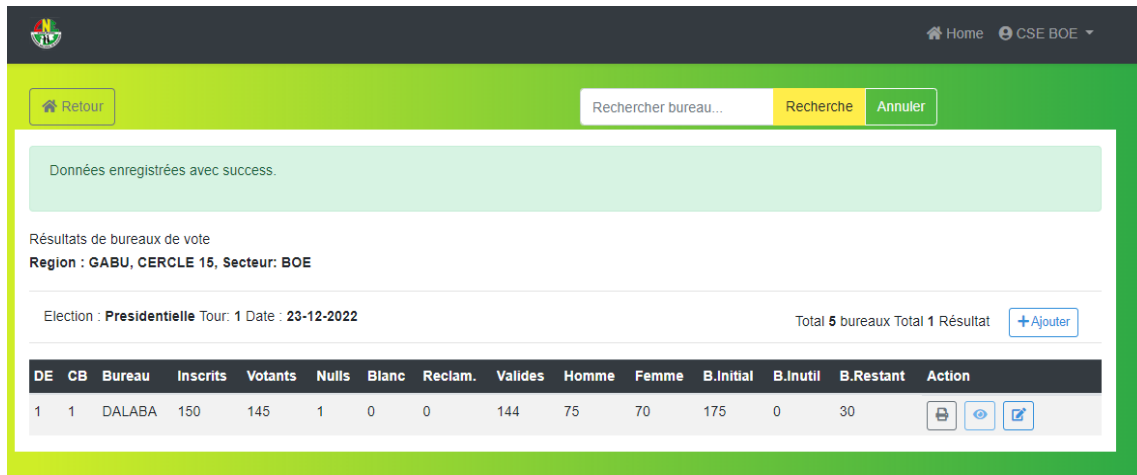


Figure 36-Apres insertion de resultat

Il peut voir, imprimer, modifier en cliquant sur les icones que se trouvent à coté sur « **Actions** ». Il peut voir les statistiques sectorielles en cliquant sur le bouton « **STATISQUE** » de barre de menu d'accueil comme le montre la figure suivante :

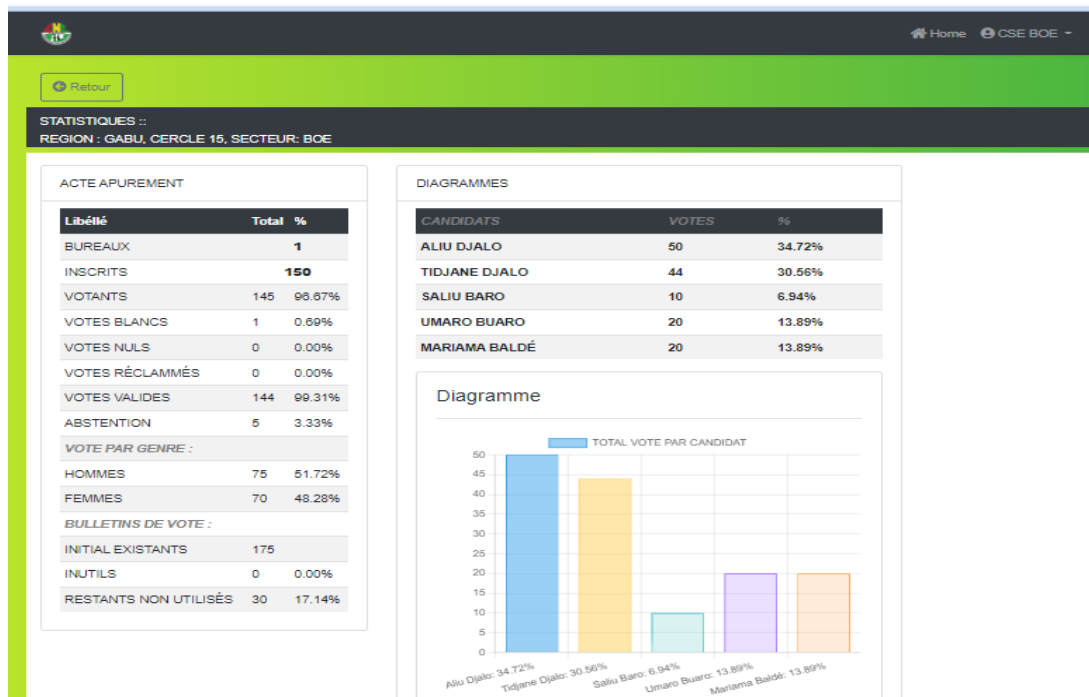


Figure 37-Statistique Sectoriel d'Election

Le délégué sectoriel peut gérer les rapports de résultats électoraux du secteur où il est affecté. Pour cela, il clique sur le bouton « **RAPPORTS** » sur la barre de menu principal. Une fois cliqué sur le bouton, voici la page de menu de gestion de rapport :

Chapitre 4 - Implémentation et Présentation de l'Application

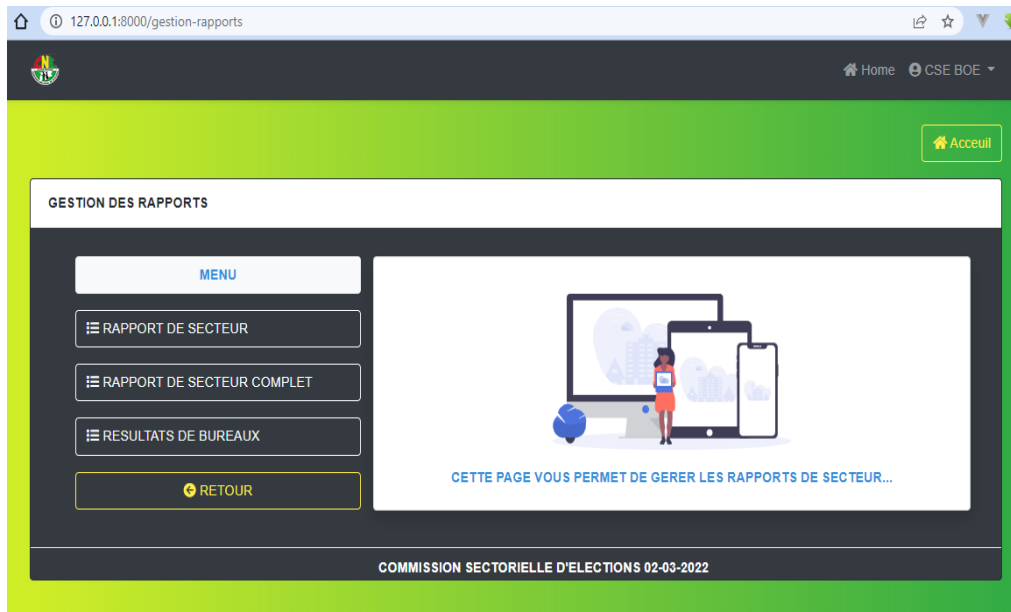


Figure 38-Menu Gestion de rapports de secteur

Le délégué peut imprimer un rapport partiel de résultat qui résume les détails en cliquant sur « **RAPPORT DE SECTEUR** » comme le montre figure suivante :

The screenshot shows a printed report from the 'REPUBLICQUE DE LA GUINE-BISSAU'. The report is titled 'RÉSULTAT PARTIEL DE VOTATION' and is for the 'ASSEMBLEE NATIONALE POPULAIRE' and 'COMMISSION NATIONAL D'ELECTION'. The election details are: 'Election: Présidentielle', 'Tour: 1', 'Date: 23-12-2022', and 'Imprimée le: 02-03-2022'. The report includes two tables. The first table shows regional and sectoral statistics, and the second table lists candidates and their vote counts.

REGION	GABU
SECTEUR	BOE
NUMERO D'INSCRITS	150
NUMERO DE VOTANTS	145
NUMERO DE VOTES BLANCS	1
NUMERO DE VOTES NULS	0
NUMERO DE VOTES RECLAMMES	0
NUMERO DE VOTES VALIDES	144
HOMMES	75
FEMMES	70

CANDIDATS :	
N° PRENOM NOM	VOTES
1 ALIU DJALO	50
2 TIDJANE DJALO	44
3 SALIU BARO	10
4 UMARO BUARO	20
5 MARIAMA BALDÉ	20

Figure 39-Rapport sectoriel résumé de résultat

Le délégué peut aussi imprimer un rapport partiel complet en cliquant sur le bouton « **Rapport de secteur complet** ».

Chapitre 4 - Implémentation et Présentation de l'Application

Le délégué peut aussi imprimer un autre rapport de bureaux de vote en cliquant sur le bouton « **RESULTATS DE BUREAU** » sur la barre de menu, comme la montre sur la figure suivante :


REPUBLIQUE DE LA GUINEE-BISSAU
ASSEMBLEE NATIONALE POPULAIRE
COMMISSION NATIONAL D'ELECTION
Résultats de Bureau de Vote

Election: **Présidentielle** Tour: **1** Date : **23-12-2022** Imprimée le: **02-03-2022**

RÉGION	CERCLE	SECTEUR	DE	CB	BUREAU	CANDIDAT	VOTE
7	15	BOE	1	1	DALABA	Aliu Djalo	50
7	15	BOE	1	1	DALABA	Tidjane Djalo	44
7	15	BOE	1	1	DALABA	Saliu Baro	10
7	15	BOE	1	1	DALABA	Umaro Buaro	20
7	15	BOE	1	1	DALABA	Mariama Baldé	20

Figure 40-Rapport résultat de bureaux de vote

4.5 Conclusion

Dans ce chapitre, nous avons procédé d'abord à l'élaboration de l'architecture applicative de notre application, ensuite aux structures de notre base de données et du projet. Et en fin nous avons procédé aux tests et validations de notre application en montrant les différentes actions que chaque utilisateur peut faire dans notre application.

CONCLUSION ET PERSPECTIVES

1. Conclusion

Nous avons réalisé, dans le cadre de ce mémoire de fin d'études, une application web à la disposition de la Commission Nationale des Élections de la Guinée Bissau. Cette dernière permettra à la CNE d'avoir un système de gestion des élections plus sécurisé et plus efficace. Elle permettra ainsi de gérer les *élections présidentielles et législatives*, de générer *automatiquement des rapports de scrutin et produire des statistiques fiables sur les résultats de bureaux*.

Pour y parvenir, nous avons d'abord effectué une étude de la gestion actuelle des élections de la CNE afin de fixer les problèmes et de se fixer des objectifs, ensuite nous avons procédé à la modélisation et à la conception de notre système avant de faire une étude des outils technologiques que nous avons utilisés. Puis, nous avons procédé à la mise en œuvre de l'application et en fin aux tests et validations.

La réalisation de ce travail nous a permis d'exploiter le vaste domaine de la conception et de la réalisation d'une application web en approfondissant nos connaissances. Ce projet a évidemment été bénéfique pour nous dans la mesure où il nous a permis de parcourir un peu tous les domaines de notre formation (en Management de Systèmes d'Informations Automatisés (MSIA)) en particulier, en développement d'applications web surtout avec le langage PHP et la gestion des bases de données.

2. Perspectives

Comme perspectives, nous envisageons d'étendre l'application en ajoutant un Datawarehouse. Un Datawarehouse (entrepôt de données en français) est une collection de données orientées sujet, intégrées, non volatiles et historiées, organisées pour le support d'un processus d'aide à la décision. [10]

Ainsi, à partir de notre base de données de production, nous allons mettre en place et alimenter un entrepôt de données qui se présente comme suit :

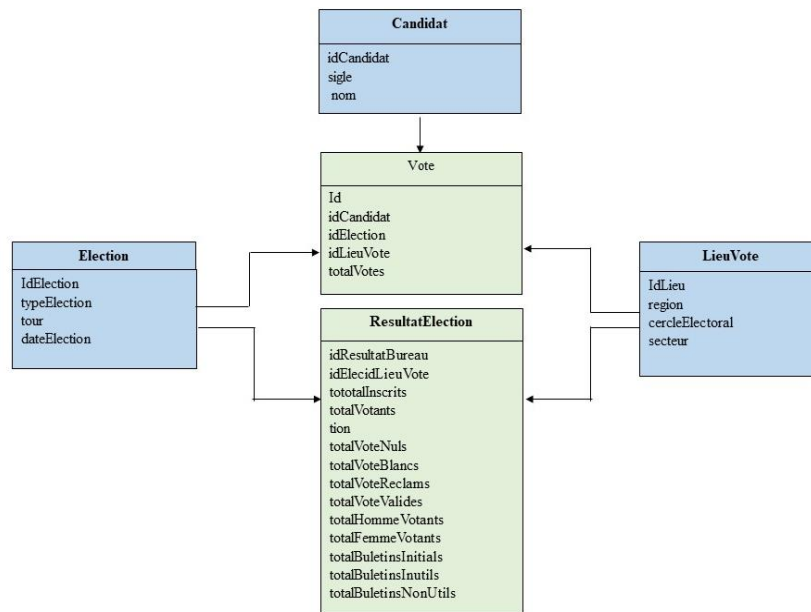


Figure 41-Datawarehouse

Cet entrepôt de données est un modèle étoile et constitué de la table de faits **Vote et Résultat** et des dimensions d'analyse à savoir :

- ✓ **Candidat** qui contient des informations sur les différents candidats ;
- ✓ **Election** contenant les informations des différentes élections ;
- ✓ **Lieu de Vote** contenant les informations de localité.

Cet entrepôt nous permettra de répondre aux questions suivantes :

- ✓ Quelle est, la région ou cercle qu'un candidat a eu plus (ou moins) votes ?
- ✓ Quels sont les candidats que n'ont jamais participé deuxième tour d'élections ?
- ✓ Quel est, pour chaque élection la région, cercle électoral, secteur, où les femmes ou les hommes votent plus (ou moins) ?
- ✓ Quel est, type d'élection que les femmes ou les hommes votent plus (ou moins) ?
- ✓ Quel est, tour d'élection ou type d'élection que les électeurs votent plus (ou moins) ?
- ✓ Quelle est, pour chaque élection la région, le cercle électoral ou le secteur, où votants est 95% de total inscrits ?
- ✓ Quelle est, l'élection qu'a plus (ou moins) les votes nuls, votes blancs ou votes réclamés ?
- ✓ Quelle est, la région, cercle électoral ou secteur qu'a-t-elle plus de votes réclamés, votes nuls ou votes blancs ?
- ✓ Quelle est, la région, cercle ou secteur a-t-il plus (ou moins) des bulletins de votes Inutiles ou bulletins de votes non utilisés ?
- ✓ Etc

BIBLIOGRAPHIE ET WEBOGRAPHIE

- [1] *Genesis*, «*Legislation Electoral (Legislação Eleitoral)*,» Bissau, *Comission Nationale d'Election*, 2019.
- [2] *P. A. Cisse*, «*Cours Analyse De Système d'Information*,» Zigunchor, *Université*, 2019.
- [3] *Demba Faye*, «*Memoire Master 2 ,Systèmes d'Information Repartis*,» Cheik Anta Diop, *Dakar /Sénégal*, 2016-2017.
- [4] *Ismail*, «<https://www.hostinger.fr/tutoriels/quest-ce-quapache-serveur-web-apache/>,» [En ligne].
- [5] «<http://laravel.com/docs>,» [En ligne].
- [6] *A. Buleté*, «<https://walkerspider.com/cours/laravel/blade>,» [En ligne].
- [7] «<http://getbootstrap.com/docs/4.0/>,» [En ligne].
- [8] *V. S. Code*, «<https://fr.wikipedia.org>,» [En ligne]. Available: https://fr.wikipedia.org/wiki/Visual_Studio_Code.
- [9] *WampServer*, «<https://fr.wikipedia.org>,» [En ligne]. Available: <https://fr.wikipedia.org/wiki/WampServer>.
- [10] *D. Sarr*, «*Cours de Systèmes d'Information Decisionels*,» 2020.
- [11] «<https://laravel.com/docs/8.x/blade#introduction>,» [En ligne].
- [12] *E. Reese*, «<https://openclassrooms.com/fr/courses/3504441-introduction-a-jquery>,» [En ligne].

ANNEXES

Annexe 1 : Fichier de configuration de la connexion .ENV

```

.. .env x
  .env
  1 APP_NAME=CNE
  2 APP_ENV=local
  3 APP_KEY=base64:jcm9lXAgnrIku/wbpps8DRIsdKNTefw9ywgGo1oZU=
  4 APP_DEBUG=true
  5 APP_URL=http://localhost
  6
  7 LOG_CHANNEL=stack
  8 LOG_LEVEL=debug
  9
 10 DB_CONNECTION=mysql
 11 DB_HOST=127.0.0.1
 12 DB_PORT=3306
 13 DB_DATABASE=dbcne
 14 DB_USERNAME=aliu
 15 DB_PASSWORD=aliu1234
 16
 17 BROADCAST_DRIVER=log
 18 CACHE_DRIVER=file
 19 QUEUE_CONNECTION=sync
 20 SESSION_DRIVER=file
 21 SESSION_LIFETIME=120
 22
 23 MEMCACHED_HOST=127.0.0.1
 24
 25 REDIS_HOST=127.0.0.1
 26 REDIS_PASSWORD=null
 27 REDIS_PORT=6379
 28
 29 MAIL_MAILER=smtp
 30 MAIL_HOST=mailhog
 31 MAIL_PORT=1025
 32 MAIL_USERNAME=null
 33 MAIL_PASSWORD=null
 34 MAIL_ENCRYPTION=null
 35 MAIL_FROM_ADDRESS=null
 36 MAIL_FROM_NAME="${APP_NAME}"
 37
 38 AWS_ACCESS_KEY_ID=
 39 AWS_SECRET_ACCESS_KEY=
 40 AWS_DEFAULT_REGION=us-east-1
 41 AWS_BUCKET=
 42
 43 PUSHER_APP_ID=
 44 PUSHER_APP_KEY=
 45 PUSHER_APP_SECRET=
 46 PUSHER_APP_CLUSTER=mt1
 47 MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
 48 MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
 49

```

Fichier .env

Annexe 2: Fichier de configuration de la base de données (database.php).

```

config > database.php
You, há 1 segundo | 1 author (You)
1  <?php
2
3  use Illuminate\Support\Str;
4
5  return [
6
7      /*
8      |-----
9      | Default Database Connection Name
10     |-----
11     */
12     'default' => env('DB_CONNECTION', 'mysql'),
13     /*
14     |-----
15     | Database Connections
16     |-----
17     */
18     'connections' => [
19
20         'sqlite' => [
21             'driver' => 'sqlite',
22             'url' => env('DATABASE_URL'),
23             'database' => env('DB_DATABASE', database_path('database.sqlite')),
24             'prefix' => '',
25             'foreign_key_constraints' => env('DB_FOREIGN_KEYS', true),
26         ],
27
28         'mysql' => [
29             'driver' => 'mysql',
30             'url' => env('DATABASE_URL'),
31             'host' => env('DB_HOST', '127.0.0.1'),
32             'port' => env('DB_PORT', '3306'),
33             'database' => env('DB_DATABASE', 'forge'),
34             'username' => env('DB_USERNAME', 'forge'),
35             'password' => env('DB_PASSWORD', ''),
36             'unix_socket' => env('DB_SOCKET', ''),
37             'charset' => 'utf8mb4',
38             'collation' => 'utf8mb4_unicode_ci',
39             'prefix' => '',
40             'prefix_indexes' => true,
41             'strict' => true,
42             'engine' => 'InnoDB',
43             'options' => extension_loaded('pdo_mysql') ? array_filter([
44                 PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
45             ]) : [],

```

Fichier database.php

Annexe 3 : Fichier composer.json

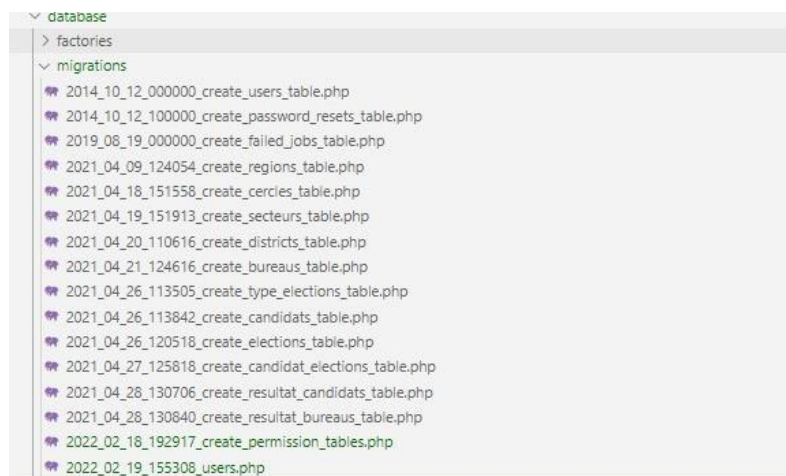
```

1  {} composer.json > ...
2  You, há 14 minutos | 1 author (You)
3  You, há 3 meses • First Commit
4  {
5  "name": "laravel/laravel",
6  "type": "project",
7  "description": "The Laravel Framework.",
8  "keywords": [
9  "framework",
10 "laravel"
11 ],
12 "license": "MIT",
13 "require": {
14 "php": "^7.3|^8.0",
15 "barryvdh/laravel-dompdf": "^0.9.0",
16 "fideloper/proxy": "^4.4",
17 "fruitcake/laravel-cors": "^2.0",
18 "guzzlehttp/guzzle": "^7.0.1",
19 "laravel/framework": "^8.12",
20 "laravel/tinker": "^2.5",
21 "laravel/ui": "^3.2",
22 "spatie/laravel-permission": "^5.5"
23 },
24 "require-dev": {
25 "facade/ignition": "^2.5",
26 "fakerphp/faker": "^1.9.1",
27 "laravel/sail": "^1.0.1",
28 "mockery/mockery": "^1.4.2",
29 "nunomaduro/collision": "^5.0",
30 "phpunit/phpunit": "^9.3.3"
31 },
32 "config": {
33 "optimize-autoloader": true,
34 "preferred-install": "dist",
35 "sort-packages": true
36 },
37 "extra": {
38 "laravel": {
39 "dont-discover": []
40 }
41 },
42 "autoload": {
43 "psr-4": {
44 "App\\": "app/",
45 "Database\\Factories\\": "database/factories/",
46 "Database\\Seeders\\": "database/seeders/"
47 }
48 },
49 "autoload-dev": {
50 "psr-4": {
51 "Tests\\": "tests/"
52 }
53 }

```

Fichier composer.json

Annexe 4 : Dossier de fichier des migrations de base de données

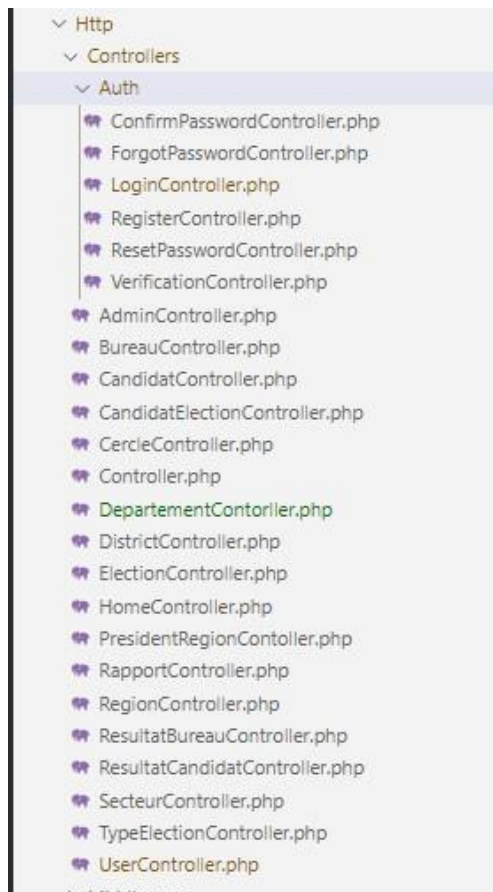


Les migrations

Annexe 5 : Les Modèles et les contrôleurs



Le dossier des Modèles



Le dossier des contrôleurs

Annexe 6 : Les routes de l'application (web.php)

```

You, há 1 segundo * Uncommitted changes
Route::middleware(['auth'])->group(function(){
    //Grupo de rotas do clientes
    Route::prefix('regions')->group(function(){
        Route::get('', [RegionController::class, 'index'])->name('region-index');
        Route::get('/create', [RegionController::class, 'create'])->name('region-create');
        Route::post('/store', [RegionController::class, 'store'])->name('region-store');
        Route::get('/{id}/edit', [RegionController::class, 'edit'])->name('region-edit');
        Route::put('/{id}/update', [RegionController::class, 'update'])->name('region-update');
        Route::delete('/{id}', [RegionController::class, 'destroy'])->name('region-destroy');
        Route::get('/pdf', [RegionController::class, 'geraPdf'])->name('print-region');
    });
    Route::prefix('cercles')->group(function(){
        Route::get('', [CercleController::class, 'index'])->name('cercle-index');
        Route::get('/create', [CercleController::class, 'create'])->name('cercle-create');
        Route::post('/store', [CercleController::class, 'store'])->name('cercle-store');
        Route::get('/{id}/edit', [CercleController::class, 'edit'])->name('cercle-edit');
        Route::put('/{id}/update', [CercleController::class, 'update'])->name('cercle-update');
        Route::delete('/{id}', [CercleController::class, 'destroy'])->name('cercle-destroy');
        Route::get('/pdf', [CercleController::class, 'geraPdf'])->name('cercles-print');
    });
    Route::prefix('secteurs')->group(function(){
        Route::get('', [SecteurController::class, 'index'])->name('secteur-index');
        Route::get('/create', [SecteurController::class, 'create'])->name('secteur-create');
        Route::post('/store', [SecteurController::class, 'store'])->name('secteur-store');
        Route::get('/{id}/edit', [SecteurController::class, 'edit'])->name('secteur-edit');
        Route::put('/{id}/update', [SecteurController::class, 'update'])->name('secteur-update');
        Route::delete('/{id}', [SecteurController::class, 'destroy'])->name('secteur-destroy');
        Route::get('/pdf', [SecteurController::class, 'geraPdf'])->name('secteur-print');
    });
    Route::prefix('districts')->group(function(){
        Route::get('', [DistrictController::class, 'index'])->name('district-index');
        Route::get('/create', [DistrictController::class, 'create'])->name('district-create');
        Route::post('/store', [DistrictController::class, 'store'])->name('district-store');
        Route::get('/{id}/edit', [DistrictController::class, 'edit'])->name('district-edit');
        Route::put('/{id}/update', [DistrictController::class, 'update'])->name('district-update');
        Route::delete('/{id}', [DistrictController::class, 'destroy'])->name('district-destroy');
        Route::get('/pdf', [DistrictController::class, 'geraPdf'])->name('district-print');
    });
}

//Les Route accessible seul pour les utilisateur authentifiées
Route::middleware(['auth'])->group(function(){
    Route::get('/commission-sectorielle-elections', [App\Http\Controllers\UserController::class, 'dashboard'])->name('home-user');

    Route::prefix('cre')->group(function(){
        // ROUTE POUR LE GESTION REGIONAL DE D'ELECTION
        Route::get('', [App\Http\Controllers\PresidentRegionContoller::class, 'homeCre'])->name('home-cre');
        Route::get('/dashboard-regional', [App\Http\Controllers\PresidentRegionContoller::class, 'dashboard'])->name('dashboard-cre');
        Route::get('/statistiques', [App\Http\Controllers\PresidentRegionContoller::class, 'statistique'])->name('rapport-dashboard-cre');
        Route::get('/rapports-regional-election', [App\Http\Controllers\PresidentRegionContoller::class, 'rapport'])->name('rapport-cre');
        Route::get('/appropos-gestion-election-regional', [App\Http\Controllers\PresidentRegionContoller::class, 'appropos'])->name('appropos-cre');
        Route::get('/rapport-regional-resume', [App\Http\Controllers\PresidentRegionContoller::class, 'rapportResumeRegionalPdf'])->name('rapport-resume-region1');
        Route::get('/rapport-regional-complet', [App\Http\Controllers\PresidentRegionContoller::class, 'rapportCompletRegionalPdf'])->name('rapport-complet-region1');

        Route::get('/cercles-index', [App\Http\Controllers\PresidentRegionContoller::class, 'rapportCercleIndex'])->name('cercles-index');
        Route::get('/{id}/{idElec}/rapport-cercles-print', [App\Http\Controllers\PresidentRegionContoller::class, 'rapportCerclePrint'])->name('cercle-print');
        Route::get('/{id}/{idElec}/rapport-cercles-show', [App\Http\Controllers\PresidentRegionContoller::class, 'rapportCercleShow'])->name('cercle-show');
        # ---- ROUTES POUR GESTION DES RAPPORTS SECTORIELE POUR COMMISSION REGIONAL D'ELECTIONS ---- #
        Route::get('/secteurs-index', [App\Http\Controllers\PresidentRegionContoller::class, 'rapportSecteurIndex'])->name('secteurs-index');
        Route::get('/{id}/{idElec}/rapport-secteur-print', [App\Http\Controllers\PresidentRegionContoller::class, 'rapportSecteurPrint'])->name('rapport-secteur-region');
        Route::get('/{id}/{idElec}/rapport-secteur-show', [App\Http\Controllers\PresidentRegionContoller::class, 'rapportSecteurShow'])->name('secteur-show');
        Route::get('/rapport-secteur-pdf', [App\Http\Controllers\PresidentRegionContoller::class, 'rapportSecteurPdf'])->name('secteur-print-pdf');

        Route::get('/rapport-resultats-bureaux-region-pdf', [App\Http\Controllers\PresidentRegionContoller::class, 'resultatCandidatBureau'])->name('rapport-regional-bureaux');
    });
    Route::prefix('cdepart')->group(function(){

```

```

160 Route::prefix('districts')->group(function(){
161     Route::get('', [DistrictController::class, 'index'])->name('district-index');
162     Route::get('/create', [DistrictController::class, 'create'])->name('district-create');
163     Route::post('/store', [DistrictController::class, 'store'])->name('district-store');
164     Route::get('/{id}/edit', [DistrictController::class, 'edit'])->name('district-edit');
165     Route::put('/{id}/update', [DistrictController::class, 'update'])->name('district-update');
166     Route::delete('/{id}', [DistrictController::class, 'destroy'])->name('district-destroy');
167     Route::get('/pdf', [DistrictController::class, 'geraPdf'])->name('district-print');
168 });
169 Route::prefix('bureaus')->group(function(){
170     Route::get('', [BureauController::class, 'index'])->name('bureau-index');
171     Route::get('/create', [BureauController::class, 'create'])->name('bureau-create');
172     Route::post('/store', [BureauController::class, 'store'])->name('bureau-store');
173     Route::get('/{id}/edit', [BureauController::class, 'edit'])->name('bureau-edit');
174     Route::put('/{id}/update', [BureauController::class, 'update'])->name('bureau-update');
175     Route::delete('/{id}', [BureauController::class, 'destroy'])->name('bureau-destroy');
176     Route::get('/pdf', [BureauController::class, 'geraPdf'])->name('bureau-print');
177 });
178 Route::prefix('typeElections')->group(function(){
179     Route::get('', [TypeElectionController::class, 'index'])->name('typeElection-index');
180     Route::get('/create', [TypeElectionController::class, 'create'])->name('typeElection-create');
181     Route::post('/store', [TypeElectionController::class, 'store'])->name('typeElection-store');
182     Route::get('/{id}/edit', [TypeElectionController::class, 'edit'])->name('typeElection-edit');
183     Route::put('/{id}/update', [TypeElectionController::class, 'update'])->name('typeElection-update');
184     Route::delete('/{id}', [TypeElectionController::class, 'destroy'])->name('typeElection-destroy');
185     Route::get('/pdf', [TypeElectionController::class, 'geraPdf'])->name('typeElection-print');
186 });
187 Route::prefix('election')->group(function(){
188     Route::get('', [ElectionController::class, 'index'])->name('election-index');
189     Route::get('/create', [ElectionController::class, 'create'])->name('election-create');
190     Route::post('/store', [ElectionController::class, 'store'])->name('election-store');
191     Route::get('/{id}/edit', [ElectionController::class, 'edit'])->name('election-edit');
192     Route::get('/{id}/show', [ElectionController::class, 'show'])->name('election-show');
193     Route::put('/{id}/update', [ElectionController::class, 'update'])->name('election-update');
194     Route::delete('/{id}', [ElectionController::class, 'destroy'])->name('election-destroy');
195     Route::get('/pdf', [ElectionController::class, 'geraPdf'])->name('election-print');
196 });
197 });
198 Route::prefix('candidat')->group(function(){
199     Route::get('', [CandidatController::class, 'index'])->name('candidat-index');
200     Route::get('/create', [CandidatController::class, 'create'])->name('candidat-create');
201     Route::post('/store', [CandidatController::class, 'store'])->name('candidat-store');
202     Route::get('/{id}/edit', [CandidatController::class, 'edit'])->name('candidat-edit');
203     Route::put('/{id}/update', [CandidatController::class, 'update'])->name('candidat-update');
204     Route::delete('/{id}', [CandidatController::class, 'destroy'])->name('candidat-destroy');
205     Route::get('/pdf', [CandidatController::class, 'geraPdf'])->name('candidat-print');
206 });
207 Route::prefix('candidatElection')->group(function(){...
208 });
209 Route::prefix('resultats-bureau')->group(function(){
210     Route::get('', [ResultatBureauController::class, 'index'])->name('rb-index');
211     Route::get('/rc', [ResultatBureauController::class, 'rc'])->name('rc-index');
212     Route::get('/create', [ResultatBureauController::class, 'create'])->name('rb-create');
213     Route::post('/store', [ResultatBureauController::class, 'store'])->name('rb-store');
214     Route::get('/{id}/{idb}/{idElec}/edit', [ResultatBureauController::class, 'edit'])->name('rb-edit');
215     Route::get('/{id}/{idb}/{idElec}/show', [ResultatBureauController::class, 'show'])->name('rb-show');
216     Route::put('/{id}/{idElec}/update', [ResultatBureauController::class, 'update'])->name('rb-update');
217     Route::delete('/{id}/{idb}/{idElec}', [ResultatBureauController::class, 'destroy'])->name('rb-destroy');
218     Route::get('/statistique', [ResultatBureauController::class, 'statistique'])->name('rb-statistique');
219     Route::get('/chart', [ResultatBureauController::class, 'chart'])->name('rb-chart');
220     Route::get('/gestion-rapport', [ResultatBureauController::class, 'gestion'])->name('rb-gRapport');
221     Route::get('/{id}/{idb}/{idElec}/pdf', [ResultatBureauController::class, 'resultatPDF'])->name('rb-print');
222 });
223 Route::prefix('result')->group(function(){
224     Route::get('', [R_BureauController::class, 'index'])->name('rbureau-index');
225     Route::get('resultC', [R_CandidatController::class, 'index'])->name('r_candidat-index');
226     Route::get('r_candidat', [R_CandidatController::class, 'c_resultat'])->name('r_candidat-c_result');
227     Route::get('/create', [insertController::class, 'index'])->name('create-result');
228     Route::post('/store', [insertController::class, 'store'])->name('store-result');
229     Route::get('/{id}/edit', [R_BureauController::class, 'edit'])->name('edit-result');
230     Route::delete('/{id}/idElec', [R_BureauController::class, 'destroy'])->name('r_bureau-destroy');
231 });
232
233

```

Annexe 7 : Schéma de la base de données

Le schéma de la base de données peut être généré automatiquement en utilisant la commande suivante :

```
mysqldump -d -u root dbcne > SchemaDataBaseCNE.sql
```

root est l'utilisateur ;

dbcne est le nom de la base de données ;

SchemaDataBaseCNE.sql est le nom du schéma après génération

```

6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
-- Table structure for table `bureaus`
--
DROP TABLE IF EXISTS `bureaus`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `bureaus` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `cod_bureau` int(11) NOT NULL,
  `bureau` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
  `district_id` bigint(20) unsigned NOT NULL,
  `secteur_id` bigint(20) unsigned NOT NULL,
  `cercle_id` bigint(20) unsigned NOT NULL,
  `region_id` bigint(20) unsigned NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `bureaus_bureau_unique` (`bureau`),
  KEY `bureaus_district_id_foreign` (`district_id`),
  KEY `bureaus_secteur_id_foreign` (`secteur_id`),
  KEY `bureaus_ceracle_id_foreign` (`ceracle_id`),
  KEY `bureaus_region_id_foreign` (`region_id`),
  CONSTRAINT `bureaus_ceracle_id_foreign` FOREIGN KEY (`ceracle_id`) REFERENCES `cercles` (`id`),
  CONSTRAINT `bureaus_district_id_foreign` FOREIGN KEY (`district_id`) REFERENCES `districts` (`id`),
  CONSTRAINT `bureaus_region_id_foreign` FOREIGN KEY (`region_id`) REFERENCES `regions` (`id`),
  CONSTRAINT `bureaus_secteur_id_foreign` FOREIGN KEY (`secteur_id`) REFERENCES `secteurs` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
-- Table structure for table `candidat_elections`
--
DROP TABLE IF EXISTS `candidat_elections`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `candidat_elections` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `cod_candidat` int(11) DEFAULT NULL,
  `candidat_id` bigint(20) unsigned NOT NULL,
  `election_id` bigint(20) unsigned NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `candidat_elections_candidat_id_foreign` (`candidat_id`),
  KEY `candidat_elections_election_id_foreign` (`election_id`),
  CONSTRAINT `candidat_elections_candidat_id_foreign` FOREIGN KEY (`candidat_id`) REFERENCES `candidats` (`id`),
  CONSTRAINT `candidat_elections_election_id_foreign` FOREIGN KEY (`election_id`) REFERENCES `elections` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=26 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Table structure for table `candidats`
--
DROP TABLE IF EXISTS `candidats`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `candidats` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `sigle` varchar(20) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `nom` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```

89 ● DROP TABLE IF EXISTS `cercles`;
90 ● /*140101 SET @saved_cs_client = @@character_set_client */;
91 ● /*140101 SET character_set_client = utf8 */;
92 ● CREATE TABLE `cercles` (
93   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
94   `cod_cercle` int(11) NOT NULL,
95   `cercle` varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL,
96   `region_id` bigint(20) unsigned NOT NULL,
97   `created_at` timestamp NULL DEFAULT NULL,
98   `updated_at` timestamp NULL DEFAULT NULL,
99   PRIMARY KEY (`id`),
100  UNIQUE KEY `cercles_cod_cercle_unique` (`cod_cercle`),
101  UNIQUE KEY `cercles_cercle_unique` (`cercle`),
102  KEY `cercles_region_id_foreign` (`region_id`),
103  CONSTRAINT `cercles_region_id_foreign` FOREIGN KEY (`region_id`) REFERENCES `regions` (`id`),
104  ) ENGINE=InnoDB AUTO_INCREMENT=31 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
105 ● /*140101 SET character_set_client = @saved_cs_client */;
106
107 --
108 -- Table structure for table `districts`
109 --
110
111 ● DROP TABLE IF EXISTS `districts`;
112 ● /*140101 SET @saved_cs_client = @@character_set_client */;
113 ● /*140101 SET character_set_client = utf8 */;
114 ● CREATE TABLE `districts` (
115   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
116   `cod_district` int(11) NOT NULL,
117   `district` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
118   `secteur_id` bigint(20) unsigned NOT NULL,
119   `cercle_id` bigint(20) unsigned NOT NULL,
120   `region_id` bigint(20) unsigned NOT NULL,
121   `created_at` timestamp NULL DEFAULT NULL,
122   `updated_at` timestamp NULL DEFAULT NULL,
123   PRIMARY KEY (`id`),
124   UNIQUE KEY `districts_district_unique` (`district`),
125   KEY `districts_secteur_id_foreign` (`secteur_id`),
126   KEY `districts_cercle_id_foreign` (`cercle_id`),
127   KEY `districts_region_id_foreign` (`region_id`),
128   CONSTRAINT `districts_cercle_id_foreign` FOREIGN KEY (`cercle_id`) REFERENCES `cercles` (`id`),
129   CONSTRAINT `districts_region_id_foreign` FOREIGN KEY (`region_id`) REFERENCES `regions` (`id`),
130   CONSTRAINT `districts_secteur_id_foreign` FOREIGN KEY (`secteur_id`) REFERENCES `secteurs` (`id`),
131  ) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
132 ● /*140101 SET character_set_client = @saved_cs_client */;

```

```

137 ● DROP TABLE IF EXISTS `elections`;
138 ● /*140101 SET @saved_cs_client = @@character_set_client */;
139 ● /*140101 SET character_set_client = utf8 */;
140 ● /*140101 SET character_set_client = utf8 */;
141 ● CREATE TABLE `elections` (
142   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
143   `tour` int(11) NOT NULL,
144   `dateElection` timestamp NOT NULL,
145   `type_election_id` bigint(20) unsigned NOT NULL,
146   `created_at` timestamp NULL DEFAULT NULL,
147   `updated_at` timestamp NULL DEFAULT NULL,
148   PRIMARY KEY (`id`),
149   KEY `elections_type_election_id_foreign` (`type_election_id`),
150   CONSTRAINT `elections_type_election_id_foreign` FOREIGN KEY (`type_election_id`) REFERENCE
151  ) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
152 ● /*140101 SET character_set_client = @saved_cs_client */;
153
154 --
155 -- Table structure for table `failed_jobs`
156 --
157
158 ● DROP TABLE IF EXISTS `failed_jobs`;
159 ● /*140101 SET @saved_cs_client = @@character_set_client */;
160 ● /*140101 SET character_set_client = utf8 */;
161 ● CREATE TABLE `failed_jobs` (
162   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
163   `uuid` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
164   `connection` text COLLATE utf8mb4_unicode_ci NOT NULL,
165   `queue` text COLLATE utf8mb4_unicode_ci NOT NULL,
166   `payload` longtext COLLATE utf8mb4_unicode_ci NOT NULL,
167   `exception` longtext COLLATE utf8mb4_unicode_ci NOT NULL,
168   `failed_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
169   PRIMARY KEY (`id`),
170   UNIQUE KEY `failed_jobs_uuid_unique` (`uuid`)
171  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
172 ● /*140101 SET character_set_client = @saved_cs_client */;
173
174 --
175 -- Table structure for table `migrations`
176 --
177
178 ● DROP TABLE IF EXISTS `migrations`;
179 ● /*140101 SET @saved_cs_client = @@character_set_client */;
180 ● /*140101 SET character_set_client = utf8 */;
181 ● CREATE TABLE `migrations` (
182   `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
183   `migration` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
184   `batch` int(11) NOT NULL,
185   PRIMARY KEY (`id`)
186  ) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
187 ● /*140101 SET character_set_client = @saved_cs_client */;

```

```

193 DROP TABLE IF EXISTS `model_has_permissions`;
194 /*!40101 SET @saved_cs_client      = @@character_set_client */;
195 /*!40101 SET character_set_client = utf8 */;
196 CREATE TABLE `model_has_permissions` (
197   `permission_id` bigint(20) unsigned NOT NULL,
198   `model_type` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
199   `model_id` bigint(20) unsigned NOT NULL,
200   PRIMARY KEY (`permission_id`,`model_id`,`model_type`),
201   KEY `model_has_permissions_model_id_model_type_index` (`model_id`,`model_type`),
202   CONSTRAINT `model_has_permissions_permission_id_foreign` FOREIGN KEY (`permission_id`) REFERENCES `permissions` (`id`) ON DELETE CASCADE
203 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
204 /*!40101 SET character_set_client = @saved_cs_client */;
205
206 --
207 -- Table structure for table `model_has_roles`
208 --
209
210 DROP TABLE IF EXISTS `model_has_roles`;
211 /*!40101 SET @saved_cs_client      = @@character_set_client */;
212 /*!40101 SET character_set_client = utf8 */;
213 CREATE TABLE `model_has_roles` (
214   `role_id` bigint(20) unsigned NOT NULL,
215   `model_type` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
216   `model_id` bigint(20) unsigned NOT NULL,
217   PRIMARY KEY (`role_id`,`model_id`,`model_type`),
218   KEY `model_has_roles_model_id_model_type_index` (`model_id`,`model_type`),
219   CONSTRAINT `model_has_roles_role_id_foreign` FOREIGN KEY (`role_id`) REFERENCES `roles` (`id`) ON DELETE CASCADE
220 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
221 /*!40101 SET character_set_client = @saved_cs_client */;
222
223 --
224 -- Table structure for table `password_resets`
225 --
226
227 DROP TABLE IF EXISTS `password_resets`;
228 /*!40101 SET @saved_cs_client      = @@character_set_client */;
229 /*!40101 SET character_set_client = utf8 */;
230 CREATE TABLE `password_resets` (
231   `email` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
232   `token` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
233   `created_at` timestamp NULL DEFAULT NULL,
234   KEY `password_resets_email_index` (`email`)
235 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
236 /*!40101 SET character_set_client = @saved_cs_client */;
237

```

```

242 DROP TABLE IF EXISTS `permissions`;
243 /*!40101 SET @saved_cs_client      = @@character_set_client */;
244 /*!40101 SET character_set_client = utf8 */;
245 CREATE TABLE `permissions` (
246   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
247   `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
248   `guard_name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
249   `created_at` timestamp NULL DEFAULT NULL,
250   `updated_at` timestamp NULL DEFAULT NULL,
251   PRIMARY KEY (`id`),
252   UNIQUE KEY `permissions_name_guard_name_unique` (`name`,`guard_name`)
253 ) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
254 /*!40101 SET character_set_client = @saved_cs_client */;
255
256 --
257 -- Table structure for table `regions`
258 --
259
260 DROP TABLE IF EXISTS `regions`;
261 /*!40101 SET @saved_cs_client      = @@character_set_client */;
262 /*!40101 SET character_set_client = utf8 */;
263 CREATE TABLE `regions` (
264   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
265   `cod_region` int(11) NOT NULL,
266   `region` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
267   `created_at` timestamp NULL DEFAULT NULL,
268   `updated_at` timestamp NULL DEFAULT NULL,
269   PRIMARY KEY (`id`),
270   UNIQUE KEY `region_UNIQUE` (`region`)
271 ) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
272 /*!40101 SET character_set_client = @saved_cs_client */;
273

```



```

277
278 ● DROP TABLE IF EXISTS `resultat_bureaus`;
279 ● /*140101 SET @saved_cs_client = @@character_set_client */;
280 ● /*140101 SET character_set_client = utf8 */;
281 ● CREATE TABLE `resultat_bureaus` (
282   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
283   `tot_inscrits` bigint(20) NOT NULL,
284   `tot_votants` bigint(20) NOT NULL,
285   `tot_vote_blanc` bigint(20) NOT NULL,
286   `tot_vote_nul` bigint(20) NOT NULL,
287   `tot_vote_recl` bigint(20) NOT NULL,
288   `tot_vote_valid` bigint(20) NOT NULL,
289   `tot_vote_homme` bigint(20) NOT NULL,
290   `tot_vote_femme` bigint(20) NOT NULL,
291   `tot_bv_initial` bigint(20) NOT NULL,
292   `tot_bv_inutil` bigint(20) NOT NULL,
293   `tot_bv_nonUtil` bigint(20) NOT NULL,
294   `election_id` bigint(20) unsigned NOT NULL,
295   `bureau_id` bigint(20) unsigned NOT NULL,
296   `district_id` bigint(20) unsigned NOT NULL,
297   `secteur_id` bigint(20) unsigned NOT NULL,
298   `cercle_id` bigint(20) unsigned NOT NULL,
299   `region_id` bigint(20) unsigned NOT NULL,
300   `created_at` timestamp NULL DEFAULT NULL,
301   `updated_at` timestamp NULL DEFAULT NULL,
302   PRIMARY KEY (`id`),
303   KEY `resultat_bureaus_bureau_id_foreign` (`bureau_id`),
304   KEY `resultat_bureaus_district_id_foreign` (`district_id`),
305   KEY `resultat_bureaus_secteur_id_foreign` (`secteur_id`),
306   KEY `resultat_bureaus_cercle_id_foreign` (`cercle_id`),
307   KEY `resultat_bureaus_region_id_foreign` (`region_id`),
308   KEY `resultat_bureaus_election_id_foreign` (`election_id`),
309   CONSTRAINT `resultat_bureaus_bureau_id_foreign` FOREIGN KEY (`bureau_id`) REFERENCES `bureaus` (`id`),
310   CONSTRAINT `resultat_bureaus_cercle_id_foreign` FOREIGN KEY (`cercle_id`) REFERENCES `cercles` (`id`),
311   CONSTRAINT `resultat_bureaus_district_id_foreign` FOREIGN KEY (`district_id`) REFERENCES `districts` (`id`),
312   CONSTRAINT `resultat_bureaus_election_id_foreign` FOREIGN KEY (`election_id`) REFERENCES `elections` (`id`),
313   CONSTRAINT `resultat_bureaus_region_id_foreign` FOREIGN KEY (`region_id`) REFERENCES `regions` (`id`),
314   CONSTRAINT `resultat_bureaus_secteur_id_foreign` FOREIGN KEY (`secteur_id`) REFERENCES `secteurs` (`id`)
315 ) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
316 ● /*140101 SET character_set_client = @saved_cs_client */;
317

```

```

324
325 ● DROP TABLE IF EXISTS `resultat_candidats`;
326 ● /*140101 SET @saved_cs_client = @@character_set_client */;
327 ● /*140101 SET character_set_client = utf8 */;
328 ● CREATE TABLE `resultat_candidats` (
329   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
330   `resultat` bigint(20) unsigned NOT NULL,
331   `candidat_election_id` bigint(20) unsigned NOT NULL,
332   `election_id` bigint(20) unsigned NOT NULL,
333   `bureau_id` bigint(20) unsigned NOT NULL,
334   `district_id` bigint(20) unsigned NOT NULL,
335   `secteur_id` bigint(20) unsigned NOT NULL,
336   `cercle_id` bigint(20) unsigned NOT NULL,
337   `region_id` bigint(20) unsigned NOT NULL,
338   `created_at` timestamp NULL DEFAULT NULL,
339   `updated_at` timestamp NULL DEFAULT NULL,
340   PRIMARY KEY (`id`),
341   KEY `resultat_candidats_candidat_election_id_foreign` (`candidat_election_id`),
342   KEY `resultat_candidats_election_id_foreign` (`election_id`),
343   KEY `resultat_candidats_bureau_id_foreign` (`bureau_id`),
344   KEY `resultat_candidats_district_id_foreign` (`district_id`),
345   KEY `resultat_candidats_secteur_id_foreign` (`secteur_id`),
346   KEY `resultat_candidats_cercle_id_foreign` (`cercle_id`),
347   KEY `resultat_candidats_region_id_foreign` (`region_id`),
348   CONSTRAINT `resultat_candidats_bureau_id_foreign` FOREIGN KEY (`bureau_id`) REFERENCES `bureaus` (`id`),
349   CONSTRAINT `resultat_candidats_candidat_election_id_foreign` FOREIGN KEY (`candidat_election_id`) REFERENCES `candidat_elections` (`id`),
350   CONSTRAINT `resultat_candidats_cercle_id_foreign` FOREIGN KEY (`cercle_id`) REFERENCES `cercles` (`id`),
351   CONSTRAINT `resultat_candidats_district_id_foreign` FOREIGN KEY (`district_id`) REFERENCES `districts` (`id`),
352   CONSTRAINT `resultat_candidats_election_id_foreign` FOREIGN KEY (`election_id`) REFERENCES `elections` (`id`),
353   CONSTRAINT `resultat_candidats_region_id_foreign` FOREIGN KEY (`region_id`) REFERENCES `regions` (`id`),
354   CONSTRAINT `resultat_candidats_secteur_id_foreign` FOREIGN KEY (`secteur_id`) REFERENCES `secteurs` (`id`)
355 ) ENGINE=InnoDB AUTO_INCREMENT=36 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
356 ● /*140101 SET character_set_client = @saved_cs_client */;
357
358 --
359 -- Table structure for table `role_has_permissions`
360 --
361 ● DROP TABLE IF EXISTS `role_has_permissions`;
362 ● /*140101 SET @saved_cs_client = @@character_set_client */;
363 ● /*140101 SET character_set_client = utf8 */;
364 ● CREATE TABLE `role_has_permissions` (
365   `permission_id` bigint(20) unsigned NOT NULL,
366   `role_id` bigint(20) unsigned NOT NULL,
367   PRIMARY KEY (`permission_id`,`role_id`),
368   KEY `role_has_permissions_role_id_foreign` (`role_id`),
369   CONSTRAINT `role_has_permissions_permission_id_foreign` FOREIGN KEY (`permission_id`) REFERENCES `permissions` (`id`) ON DELETE CASCADE,
370   CONSTRAINT `role_has_permissions_role_id_foreign` FOREIGN KEY (`role_id`) REFERENCES `roles` (`id`) ON DELETE CASCADE
371 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
372 ● /*140101 SET character_set_client = @saved_cs_client */;
373

```

```

376 DROP TABLE IF EXISTS `roles`;
377 /*140101 SET @saved_cs_client = @@character_set_client */;
378 /*140101 SET character_set_client = utf8 */;
379 CREATE TABLE `roles` (
380   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
381   `name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
382   `guard_name` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
383   `created_at` timestamp NULL DEFAULT NULL,
384   `updated_at` timestamp NULL DEFAULT NULL,
385   PRIMARY KEY (`id`),
386   UNIQUE KEY `roles_name_guard_name_unique` (`name`,`guard_name`)
387 ) ENGINE=InnoDB AUTO_INCREMENT=38 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
388 /*140101 SET character_set_client = @saved_cs_client */;
389
390 --
391 -- Table structure for table `secteurs`
392 --
393
394 DROP TABLE IF EXISTS `secteurs`;
395 /*140101 SET @saved_cs_client = @@character_set_client */;
396 /*140101 SET character_set_client = utf8 */;
397 CREATE TABLE `secteurs` (
398   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
399   `cod_secteur` int(11) NOT NULL,
400   `secteur` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
401   `cercle_id` bigint(20) unsigned NOT NULL,
402   `region_id` bigint(20) unsigned NOT NULL,
403   `created_at` timestamp NULL DEFAULT NULL,
404   `updated_at` timestamp NULL DEFAULT NULL,
405   PRIMARY KEY (`id`),
406   UNIQUE KEY `secteurs_secteur_unique` (`secteur`),
407   KEY `secteurs_cercle_id_foreign` (`cercle_id`),
408   KEY `secteurs_region_id_foreign` (`region_id`),
409   CONSTRAINT `secteurs_cercle_id_foreign` FOREIGN KEY (`cercle_id`) REFERENCES `cercles` (`id`),
410   CONSTRAINT `secteurs_region_id_foreign` FOREIGN KEY (`region_id`) REFERENCES `regions` (`id`)
411 ) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
412 /*140101 SET character_set_client = @saved_cs_client */;

```

```

416 --
417
418 DROP TABLE IF EXISTS `type_elections`;
419 /*140101 SET @saved_cs_client = @@character_set_client */;
420 /*140101 SET character_set_client = utf8 */;
421 CREATE TABLE `type_elections` (
422   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
423   `typeElection` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
424   `created_at` timestamp NULL DEFAULT NULL,
425   `updated_at` timestamp NULL DEFAULT NULL,
426   PRIMARY KEY (`id`),
427   UNIQUE KEY `type_elections_typeelection_unique` (`typeElection`)
428 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
429 /*140101 SET character_set_client = @saved_cs_client */;
430
431 --
432 -- Table structure for table `users`
433 --
434
435 DROP TABLE IF EXISTS `users`;
436 /*140101 SET @saved_cs_client = @@character_set_client */;
437 /*140101 SET character_set_client = utf8 */;
438 CREATE TABLE `users` (
439   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
440   `name` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
441   `email` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
442   `profil` varchar(191) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
443   `email_verified_at` timestamp NULL DEFAULT NULL,
444   `password` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
445   `secteur_id` bigint(20) unsigned DEFAULT NULL,
446   `cercle_id` bigint(20) unsigned DEFAULT NULL,
447   `region_id` bigint(20) unsigned DEFAULT NULL,
448   `remember_token` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
449   `created_at` timestamp NULL DEFAULT NULL,
450   `updated_at` timestamp NULL DEFAULT NULL,
451   `is_active` tinyint(1) NOT NULL DEFAULT '0',
452   PRIMARY KEY (`id`),
453   UNIQUE KEY `email` (`email`)
454 ) ENGINE=InnoDB AUTO_INCREMENT=34 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
455 /*140101 SET character_set_client = @saved_cs_client */;
456 /*140103 SET TIME_ZONE=@OLD_TIME_ZONE */;
457
458 /*140101 SET SQL_MODE=@OLD_SQL_MODE */;
459 /*140014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
460 /*140014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
461 /*140101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
462 /*140101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
463 /*140101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
464 /*140111 SET SQL_NOTES=@OLD_SQL_NOTES */;
465
466 -- Dump completed on 2022-03-21 21:08:09
467

```

Annexe 8 : La Commission Nationale Des Élections

La Commission Nationale des Élections (CNE)

ARTICLE 1

Nature Et Finalité

1. La Commission Nationale des Élections, désigné **CNE**, est un organe indépendant et permanent qui collabore avec l'Assemblée Nationale Populaire et à la fonction de superintendance, d'organisation et de gestion de processus électoral et référendaire ⁹
2. La CNE est l'unique organe pour organiser et gérer les Élections Présidentielles, Législatives et Autarchiques¹⁰.

La Commission Régionale des Élections (CRE)

ARTICLE 19

Composition et Dénomination

1. Les CRE sont constituées par :

- a) Un président, choisi par les membres de Secrétariat Exécutif de la CNE.
 - b) Un représentant pour chaque parti ou coalition de partis concurrents.
2. Les candidats indépendants pour les élections présidentielles peuvent désigner un représentant à la CRE.
 3. Les partis politiques ou coalition des partis politiques peuvent désigner un représentant à la CRE, seulement si les candidats n'ont pas le désigné.
 4. Les membres des CRE sont nommés et assermentés par le président de la CNE.

A. La Composition des bureaux (Assemblées) de vote

ARTICLE 50

Bureaux (Assemblée) De Vote

⁹ Article 1° (Nature et fins) de décret N° 12/2013, Loi de la Commission Nationale des Élections

¹⁰ Article 1° (Nature et fins) de décret N° 12/2013, Loi de la Commission Nationale des Élections

1. Les bureaux de vote sont composés d'environ quatre cents électeurs par réunion et doit coïncider avec la division établie pour l'inscription des électeurs.
2. Le plan définitif des bureaux de vote doit être divulgué par la CNE, par le biais des médias et affichées dans tout lieu public ou facilement accessible au public, trente jours avant les élections.
3. La CNE peut créer jusqu'à 8 jours avant le début des opérations de vote, les bureaux de vote dans n'importe quelle partie du pays, que doit être affichés et diffusés dans les lieux qui le composent.
4. Le tableau du bureau de vote doit de préférence inclure les éléments des brigades de recensement de la zone concernée.

Vote

ARTICLE 69

Début De Vote

1. Il appartient aux présidents des bureaux de vote de déclarer l'ouverture du scrutin.
2. Le vote commence à 7 heures du matin le jour prévu pour les élections, après constitution des membres de bureau de vote, et se termine à 17 heures.
3. Avant le début du vote, les présidents des bureaux de vote procèdent avec les autres membres des bureaux et les délégués de la liste, la vérification de la cabine de vote, des documents de travail de la table et afficher, devant les présidents, les urnes des votes pour assurer qu'elles sont vides.
4. S'il n'y a pas d'irrégularité, votent les présidents, les secrétaires, scrutateurs et délégués de la liste.

ARTICLE 77

Vote Blanc Et Nuls

1. Correspond à un vote blanc, le cas où sur le bulletin de vote il n'y a pas eu aucune marque.
2. Correspond à un vote nul, le bulletin de vote dans lequel :
 - a) Plus d'un carré a été coché ou lorsqu'il y a un doute sur lequel carré marquer ;
 - b) La case correspondant à un candidat qui s'est retiré a été cochée des élections.
 - c) A été trouvé dans l'isoloir à l'extérieur de l'urne ;

3. Il n'est pas considéré comme nul le vote correspondant au bulletin de vote dans lequel le signe X, bien qu'il ne soit pas parfaitement dessiné ou dépasse les limites du carré, vérifiez incontestablement la volonté de l'électeur.

ARTICLE 78

Doutes, Réclamations, Protestations Et Contre-Manifestations

1. Hormis les délégués des candidats, tout électeur présent au bureau de vote peut soulever des doutes et soumettre des réclamations, protestations et contre-manifestations connexes les opérations électorales de la même assemblée et les instruire avec les documents appropriés.
2. Le président de bureau ne peut pas refuser de recevoir des plaintes, protestations et contre-protestations et doit les parapher et les joindre à l'acte d'apurement.
3. Les Réclamations, protestations et contre-protestations qui doivent faire l'objet d'une délibération par la table, peuvent attendre la fin du vote, s'il est estimé que cela n'affecte pas le déroulement normal du vote.
4. Toutes les délibérations du bureau sont prises à la majorité des membres présents et justifiée, le vote du président étant prépondérante.

Les évaluations de vote

a) L'évaluation locale

ARTICLE 79

Opérations Préliminaires

Le président du bureau de vote sépare les bulletins de vote qui n'étaient pas utilisés et ceux qui, avec cette indication, étaient inutilisables, les plaçant dans des enveloppes séparées dûment paraphées et cachetées et verrouiller la liste des électeurs, qui sera signé par tous les membres du bureau et la liste des délégués présents.

ARTICLE 80

L'Ouverture Des Urnes

1. Une fois le scrutin clos, le président de séance ouvre l'urne, et suit l'opération de comptage, afin de vérifier la correspondance entre les numéros de bulletin d'urnes et le nombre d'électeurs qui ont voté à ce bureau de vote, en présence des autres membres.

2. Si le nombre de bulletins de vote dans l'urne est inférieur au nombre d'électeurs, le nombre de bulletins de vote dans l'urne est valable aux fins de dépouillement.
3. Lorsque le nombre de bulletins dans l'urne est supérieur au nombre de votants, le vote sera répété à cette table dans les 48 heures.

ARTICLE 81

Comptage

1. Le dépouillement des bulletins de vote se fait comme suit :
 1. Le président ouvre l'urne, en présence des autres membres ;
 2. Le premier scrutateur note les votes attribués à chaque liste ou candidat sur une feuille de papier blanc ou s'il existe sur un grand tableau ;
 3. Le deuxième scrutateur place, séparément et par lots après les avoir affichés, les votes déjà lus correspondant à chacune des listes, les votes blancs et les votes nuls ;
 4. Le premier et le troisième scrutateur comptent les votes et le Président à la divulgation du nombre de votes qui correspondent à chaque liste ou candidat.
1. A l'issue de l'opération visée au numéro précédent, le président du bureau procède à la confrontation entre le nombre de votes existant dans l'urne et le nombre de votes pour chaque lot.
2. Les délégués de liste ont le droit de vérifier si le dépouillement des bulletins de vote, sans toutefois modifier l'ordre de disposition des bulletins de vote, et peut se réclamer, en cas de doute, au président du bureau qui analyse la réclamation.
3. Si la réclamation n'est pas traitée par les membres de bureau, le bulletin de vote en question est placé séparément pour l'application des dispositions du numéro deux de l'article quatre-vingt de la présente loi.

ARTICLE 83

Destination Des Bulletins De Vote

1. Les votes nuls et les bulletins de vote qui n'ont pas été utilisés et ceux qui sont inutiles sont signés par le président de bureau et par les délégués de la liste et placé dans deux enveloppes qui, après avoir été dûment scellés, seront transmis à la CRE.
2. Les votes qui ont fait l'objet de réclamation sont signés par le président de bureau et par les délégués de la liste qui ont réclamés et placés dans une enveloppe qui, après avoir été correctement scellée, il est transmis à la CRE.

3. Les bulletins de vote valablement déposés sont placés dans des enveloppes scellées et transmis à la CRE qui, dans les 48 heures suivant la publication des résultats définitifs, sont transmis à la CNE pour qu'au bout d'un an, il promeuve sa destruction.

b) L'évaluation de cercle

ARTICLE 83

Opération Générale De L'évaluation Du Cercle

L'opération de dépollution par cercle consiste à :

- a) Vérifier le nombre total d'électeurs votants dans la circonscription ;
- b) Vérifier le nombre total de votes obtenus par chaque candidat, parti ou coalition de partis.
- c) Vérifier et comparer, avec la présence obligatoire des délégués des listes, sous la supervision des présidents de bureau et la fiscalisation des délégués du ministère public, tous les résultats de vote dans les bureaux de vote qui composent le cercle.

ARTICLE 84

Publication Des Résultats

Les résultats obtenus par cercle sont annoncés par le Président de la CNE puis publié par les affiches au siège de cercle électoral et diffusés par les médias.

ARTICLE 85

Rapport de l'évaluation générale des cercles

Rapport de l'évaluation générale des cercles

1. Des opérations de calcul de résultat par cercle, les minutes sont immédiatement dressées là où elles figurent les résultats obtenus, les réclamations, protestations et contre-manifestations présentées et les décisions qui ont été prises à leur sujet.
2. Deux exemplaires de chaque rapport de l'évaluation générale du cercle sont envoyés par le Président de la CRE à la CNE dans les 24 heures suivant la fin de l'évaluation général par cercle.
3. Le troisième exemplaire du rapport et tous les documents des opérations électorales qui, de force de cette loi, ils n'ont pas à remonter à la CNE, ils sont remis au Gouverneur de Région qui garde sous sa garde et sa responsabilité.

c) Évaluation Régionale

ARTICLE 86

Entité Compétente

Au niveau des Régions, l'évaluation des résultats électoraux est réalisée par la CRE en présence des délégués sur la liste après avoir centralisé les résultats électoraux obtenus dans tous les bureaux de vote établis dans les limites géographiques de leur juridiction.

ARTICLE 88

Opération De L'évaluation Régionale

L'opération de l'évaluation régionale consiste :

- a) En vérifiant le nombre total d'électeurs votants dans la Région ;
- b) En vérifiant le nombre total de votes obtenues par chaque candidat, parti politique ou coalition de partis et le nombre de votes nuls.

ARTICLE 89

Le Rapport De L'évaluation Régional

1. Les rapports des opérations de l'évaluation régionales sont établis, contenant les résultats calculés, les réclamations, protestations et contre-manifestations présentées et les décisions prises concernant le sujet.
2. Pour les besoins du numéro précédent, la CNE crée son propre modèle.
3. Le président de la CRE doit, dans les 24 heures, transmettre à la CNE deux exemplaires du rapport de l'évaluation régionale.
4. Le troisième exemplaire du rapport et tous les documents des opérations électorales qui, par force de la présente Loi, ne doivent pas remonter à la CNE, délivrés à la CRE.

d) L'évaluation Nationale

ARTICLE 90

Entité compétente

La CNE est chargée de centraliser les résultats obtenus dans chaque Région, l'évaluation et la diffusion des résultats généraux des élections et la diffusion des mandats.

ARTICLE 93

Opération de l'évaluation nationale

Le fonctionnement de l'évaluation nationale consiste à :

- a) Vérifier le nombre total d'électeurs inscrits, les électeurs qui ont voté et leur pourcentage par rapport au premier ;
- b) Vérifier le nombre total de voix obtenues par chaque candidat, parti ou coalition des partis et le nombre de votes nuls ;
- c) Dans la répartition des mandats des députés conformément aux dispositions de la présente loi et, dans la proclamation du candidat présidentiel élu.
- d) Pour déterminer les candidats élus par chaque parti ou coalition des partis.

ARTICLE 94

Publications des résultats nationaux

Entre 7 et 10 jours après la date de clôture des votes, la CNE annonce les résultats de l'évaluation nationale dans les différents médias, fixée par avis à la porte de leurs locaux.