

*République Démocratique du Congo*  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET UNIVERSITAIRE  
**UNIVERSITE WILLIAM BOOTH**



**FACULTÉ DES SCIENCES INFORMATIQUES ET  
INTELLIGENCE ARTIFICIELLE**

**MISE EN PLACE D'UN SYSTÈME DE VOTE MOBILE  
POUR LES ELECTIONS PRESIDENTIELLES.  
« CAS DE LA COMMISSION ELECTORALE NATIONALE  
INDEPENDANTE. »**



**KONGA MASSAMBA Théophile**



**San Moi**

Travail de fin d'étude présenté et défendu  
en vue de l'obtention du titre de Licencié  
en Sciences Informatiques et Intelligence  
Artificielle.

Option: Conception des systèmes d'informations

Directeur: KUTANGILA MAYOYA David  
Professeur

Rapporteur: C.T MALANDA LANDU KUYI Anderson  
Chargé d'Enseignement

**Année Académique 2023-2024**

[i]

## **EPIGRAPHIE**

« L'Informatique est le moyen de transport plus rapide pour arriver à la destination. »

**Bertin MAZUNZE**

## **DEDICACE**

A l'Eternel Dieu tout puissant créateur de l'Univers ;

A vous mes parents : MALADIE MAGUEMA Sylvain et MBUDI Pauline pour votre amour, les sacrifices consentis envers nous ;

A vous mon grand frère Ferros MAFUALA MASSAMBA, merci pour le soutien moral, financier et spirituel jusqu'à l'aboutissement de nos études du premier cycle.

Veillez trouver ici, l'expression de notre profonde gratitude.

**KONGA MASSAMBA Théophile**

## REMERCIEMENTS

Nous remercions infiniment le Dieu de toute providence qui nous à accorder sa grâce d'atteindre ce niveau d'étude, du début de l'année académique, il nous a procuré une bonne santé, sagesse et intelligence et voici qu'il a acheminé avec nous jusqu'à la fin malgré les difficultés rencontrées.

Nos remerciements s'adressent aux autorités académiques et au corps professoral de l'Université William Booth « U.W.B » en générale, et à ceux de la faculté Gestion Informatique en particulier, qu'ils n'ont cessé de donner le meilleur d'eux pour nous instruire.

Nous remercions particulièrement notre directeur le Professeur KUTANGILA MAYOYA David et à notre lecteur l'assistant Anderson MALANDA KUYI pour leurs multiples conseils qui nous ont permis de rédiger avec beaucoup de finesse le présent travail.

Nous n'oublions pas à remercier mes parent MALADIE MAGUEMA Sylvain et MBUDI Pauline.

A nos frères et sœurs : Ferros MASAMBA, Fabrice MASSAMBA, Maxime NUNI, Jurjitha, Kerlin NGONDA, Djulia MAKAYA, Annetie KITENGE.

A toutes les Familles MALADE et MASSAMBA en RDC ou à l'étranger.

Nous ne saurons terminer ces propos sans remercier à toute la famille de l'UWB et nos condisciples nous vous disons un grand merci : Esther MUNEMA, Jordan MUKOKO, Grace LELO, Nathan KANKOLONGO, Jaks MATADI, Ketsia.

**KONGA MASSAMBA Théophile**

## **0. INTRODUCTION GENERALE**

Les élections constituent un des éléments clés des systèmes démocratiques. Elles sont pourtant de plus en plus boudées par les citoyens qui ne se déplacent véritablement en masse que lorsque les enjeux leur apparaissent très clairement.

Dans ce contexte, la mise en place de modalités alternatives de vote est souvent perçue comme un renouvellement possible de la relation politique tant en terme de participation des citoyens que de symbolique républicaine.

L'institution de la démocratie comme régime politique dans nos cités a conduit à une multiplication du nombre de consultations électorales : on vote pour tous et partout. Il faut noter que, le coût d'une élection reste très élevé; cependant en dépit de l'augmentation croissante du nombre de consultations électorales, le taux de participation aux échéances électorales baisse de plus en plus pour plusieurs raisons parmi lesquelles : les longues files d'attentes observées devant les bureaux de vote et la fraude électorale qui contribue au désintérêt d'une partie non-négligeable de la population et il est désormais possible de diminuer ce coût et de rendre le processus de vote moins contraignant grâce aux protocoles de votes mobile.

De nos jours, les nouvelles technologies de l'informatique et de la communication ne cessent de domestiquer la plupart des domaines traditionnellement entretenus. Or, les élections ont à ce jour peu bénéficiées d'outils modernes pour faciliter le déroulement, améliorer la fiabilité, la manipulation, la vitesse, la divulgation et la précision des résultats.

Cependant, la situation géographique des différents bureaux de vote et l'embrouille quant au choix des bureaux de vote restent un frein

pour certaines personnes. Toutefois, Comment remédier à tous ces problèmes ? C'est pour limiter les nombreuses irrégularités du système de vote électronique que nous avons choisi de mettre sur pied un « Système de Vote mobile ».

Depuis quelques années, les téléphones intelligents (Smartphones) ont commencé à imposer leur forte présence grâce au progrès de la technologie qui les a rendus plus petits et plus efficaces et intelligents. En effet, on remarque que les utilisateurs de ce genre de téléphone deviennent de plus en plus nombreux, d'où, avoir un Smartphone est devenu une nécessité quasi-incontournable. Et suite à cette popularité croissante, on trouve aujourd'hui que le développement mobile serait en phase de devenir le domaine informatique le plus demandé dans un avenir très proche.

## **1. CHOIX ET INTERET DU SUJET**

### **1.1. CHOIX DU SUJET**

Notre choix a été porté sur ce sujet dans le but d'innover le Système de vote de la CENI, leurs permettre de rattraper les innovations de la technologie. D'aidé la CENI pour une élection libre, crédible et transparente.

### **1.2. INTERET DU SUJET**

Pour la réalisation de notre Travail de Fin d'Etude du deuxième cycle de Licence en sciences informatique et Intelligences Artificielles, le système de vote a retenu notre attention ainsi, le sujet de notre réflexion scientifique se présente comme suit : « **MISE EN PLACE D'UN SYSTEME DE VOTE MOBILE DES ELECTIONS PRESIDENTIELLES** ». Cas de la Commission Electorale Nationale Indépendante R.D. Congo, CENI en sigle.

Nous avons porté notre choix et nous nous sommes intéressés pour mettre en place un système de vote mobile afin que la CENI puisse

maximisée le nombre des votants, et la diminution du coût des achats de matériels.

Notre projet consiste à développer une application mobile (Hybride) et d'exploiter une nouvelle technologie mobile (Hybride) qui se base de l'utilisation des systèmes d'exploitation androïde, IOS, Windows, etc... destinée aux tablettes tactiles et Smartphones sous Androïde, IOS, Windows, etc... Et dans notre application, nous avons essayé d'intégrer le concept du « m-voting » dans le terme de programmation en concertant et réalisant une application pour le vote mobile et pour atteindre nos objectifs, nous avons utilisé la technologie Hybride.

## **2. DE LIMITATION DU SUJET**

### **2.1. DANS LE TEMPS**

Spécialement, notre travail couvre la période allant de 2023 à nos jours.

### **2.2. DANS L'ESPACE**

Temporairement notre champ d'investigation demeure à la Commission Electorale Nationale Indépendante, précisément pour les élections présidentielles dont l'implantation est sur Boulevard du 30 juin, avenue Isiro, à Kinshasa/ Gombe.

## **3. SUBDIVISION DU TRAVAIL**

**Notre travail sera constitué de trois chapitres :**

Chapitre 1 : Spécification et analyse des besoins.

Chapitre 2 : Etape Conceptuelle

Chapitre 3 : Implémentation et Présentation de l'application

## **CHAPITRE I SPECIFICATION ET ANALYSE DES BESOINS**

### **I. Introduction :**

Dans ce chapitre, nous mettons le vote mobile dans son cadre général. Par la suite, nous abordons l'étude de l'existant de notre travail, suivie d'une critique pour pouvoir dégager les contraintes à respecter pendant la réalisation de notre travail.

Ainsi, ce chapitre présente l'ensemble des besoins qu'ils soient fonctionnels et non fonctionnels.

Le vote est le moyen d'expression central d'une démocratie. Il existe dans toutes les sphères du pays au sein des associations et des entreprises, dans les universités et, bien évidemment, au niveau politique. Le développement des technologies de l'information semble pouvoir modifier la physionomie et le déroulement du vote.

Depuis de nombreuses années le vote électronique existe sous la forme de machine à voter, le développement d'internet ouvre un nouveau champ d'expression du suffrage en permettant le vote à distance.

### **I.1. Etude de l'existant :**

#### **I.1.1. Historique**

La Commission électorale nationale indépendante (CENI), appelée CEI jusqu'en mars 2011, a pour mandat « de garantir des élections libres et démocratiques ». Composée de 21 membres, elle est présidée par l'abbé Malu Malu. Elle organise en premier lieu le référendum constitutionnel les 18 et 19 décembre 2005.

#### **➤ Présidents**

- l'abbé Apollinaire Malu Malu, 2003-février 2011
- le pasteur Daniel Ngoyi Mulunda, février 2011-juin 2013
- l'abbé Apollinaire Malu Malu, (juin 2013-octobre 2015)
- Corneille Nangaa, (2015-2021)
- Denis Kadima, depuis 2021



### **I.1.2. SITUATION GEOGRAPHIQUE**

La Commission Electorale Nationale Indépendante, précisément pour les élections présidentielles dont l'implantation est sur Boulevard du 30 juin, avenue Isiro, à Kinshasa/ Gombe.

### **I.1.3. Mission**

La Commission électorale nationale indépendante (CENI) a pour missions :

- La constitution, la gestion et la conservation du fichier électoral national ;
- L'organisation et la supervision des opérations électorales et référendaires.

### **I.1.4. Moyen humain (Services)**

**PROTO** : Service du Protocole

**SP** : Secrétaire Particulier

**C.S** : Cellule de Sécurité

**SIFE** : Service de l'Informatique et du Ficher Electoral

**SE** : Service Exploitation

**SLME** : Service de la Formation et de la Documentation Electorale

**DOE** : Direction des Opérations Electorales

**DAF** : Direction de l'Administration et des Finances

**DMP** : Directions de Marchés Publics

**DRH** : Direction des Ressources Humaines

**DCEC** : Direction de la Communication et de l'Education Citoyenne

**B.E** : Bureau d'Etudes

**SCC** : Service Central du Courrier

**SDA** : Service de la Documentation et des Archives

**SEPB** : Service de la Programmation et de l'Exécution Budgétaire

**SCP** : Service de la Commande Publique

**SM** : Service des Matières

**SMT/PI** : Service des Marchés de Travaux et de Prestations Intellectuelles

**SMF/SC** : Service des Marchés de Fournitures et de Services Courantes

**SGCFP** : Service de la Gestion des Carrières et de la Formation Professionnelles

**SGAAS** : Service de la Gestion Administrative et Activités Sociales

**SCD** : Service Chargé des Démembrements

**SCRP** : Service de la Communication et des Relations Publics

**SEC** : Service Chargé d'Education Citoyenne

**STRP** : Service Technique Chargé de Réalisation et des Productions

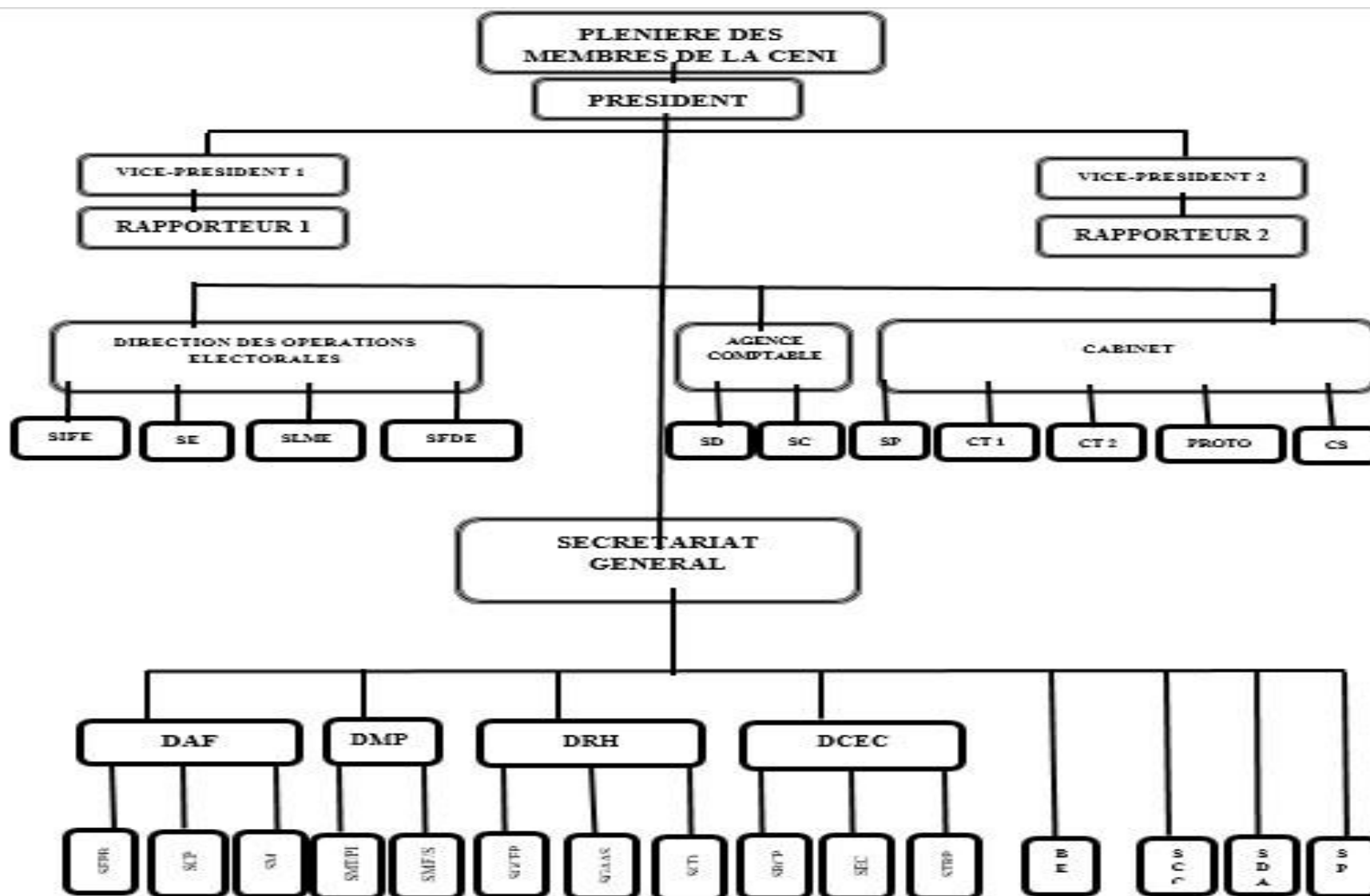
**SC** : Service de la Comptabilité

**SD** : Service de la Dépense

#### **I.1.5. Moyen matériel**

<b>Désignation</b>	<b>Nombre</b>	<b>Etat</b>
Ordinateurs fixes	103	Assez bon
Ordinateurs Portables	206	Assez bon
Imprimantes Multifonctions	27	Assez bonne
Serveurs	4	Assez bon
Machines à Voter	26.000	Assez bonne

## I.1.6. Organigramme de la CENI



## **1.2. Description de l'existant :**

En République Démocratique du Congo le système de vote jusqu'à aujourd'hui se déroule de façon électronique avec l'utilisation de machines à voter il y a quatre types d'élection :

1. L'élection présidentielle ;
2. L'élection législative Nationale ;
3. L'élection législative provinciale ;
4. L'élection Municipale.

Nous avons choisi de travailler sur le 1<sup>er</sup> type d'élection « élection présidentielle ». Les élections présidentielles ont lieu tous les cinq ans pour élire le président de la république au suffrage direct et général.

### **Il y a deux méthodes pour voter :**

Processus de jour « J » : Le jour « J », l'électeur se présente au bureau de vote muni de sa pièce d'identité (carte d'électeur) pour accomplir son acte de citoyenneté et de responsabilité, il est mis à la disposition de l'électeur, au niveau du bureau de vote, une enveloppe et des bulletins de vote (un bulletin pour chaque liste de candidats) pour voter.

L'électeur se présente à l'isoloir pour exprimer son choix en introduisant un seul bulletin de vote dans la machine à voter. L'enveloppe et l'introduit par la suite dans l'urne.

### **I.2.1. Historique du vote électronique**

Les scanners sont utilisés pour la première fois en 1962 à Kern City, en Californie. Ils constituent la plus ancienne forme de vote électronique. Les électeurs s'expriment en cochant les cases de leur choix sur le bulletin habituel. À la fin du vote, les bulletins nourrissent un scanner qui fournit le décompte des suffrages. Les ordinateurs de vote apparaissent en 1975 aux Pays-Bas et se répandent en Inde, au Brésil, en Belgique, aux États-Unis, et marginalement en France à partir de 2004. Pour exprimer leur choix, les

électeurs interagissent directement avec le dispositif via des boutons physiques, un écran tactile, ou encore un pavé numérique. Les suffrages sont enregistrés directement dans une mémoire. À la clôture de la période de vote, le dispositif fournit les résultats électoraux par affichage ou impression d'un ticket. Certains ordinateurs de vote transmettent également leurs résultats vers le bureau centralisateur. Le concept d'ordinateur de vote avec trace papier émerge en 2000 afin de pallier l'impossibilité de vérifier les résultats émis par un ordinateur de vote : lorsque l'électeur a confirmé les choix qu'il a émis, l'ordinateur imprime un ticket portant les choix exprimés. Ce bulletin, dont l'électeur peut vérifier la justesse, est ensuite dirigé vers une urne à des fins d'éventuelles vérifications des résultats. Depuis donc les années 2000, les systèmes de vote ont connu pas mal d'évolution. Néanmoins, chaque système fonctionne selon un environnement contrôlé ou non contrôlé.

## **Les environnements contrôlés et non contrôlés**

### **a) Les environnements contrôlés**

Les systèmes conçus dans ce type d'environnement sont contrôlés. Le système de vote est fixe et cela requiert la présence physique de l'électeur. Le processus de vote et le choix de l'électeur sont enregistrés sur un serveur central sécurisé fonctionnant sur un réseau privé qui est difficile à attaquer. Le choix de l'électeur ne peut donc être dévié par des tiers malintentionnés. Ces systèmes sont des ordinateurs fixes présents dans les bureaux de vote et qui gèrent tous les processus de vote.

## **Les problèmes**

- Par procuration.
- Et dans n'importe quel cas des cas cités ci-dessus nous allons dire que le bulletin est nul :
- L'enveloppe sans bulletin ou le bulletin sans enveloppe.
- Plusieurs bulletins dans une enveloppe.
- Les enveloppes ou bulletins comportant des mentions, griffonnés ou déchirés.
- Les bulletins entièrement ou partiellement barrés.
- Les bulletins ou enveloppes non réglementaires.

### **1.2.2. Critique de l'existant :**

#### **1.2.2.1. Problématique**

Les problèmes du système actuel de vote électronique avec de machine à voter comprennent entre autres ce qui suit :

- Le processus est long et couteux : Le processus de collecte de données et la saisie des données prennent trop de temps, par exemple, il y a beaucoup de temps et d'argent sont dépensés pour imprimer les formulaires de saisie de données, le temps pour préparer de stations d'enregistrement ainsi que les ressources humaines, y compris la sensibilisation des électeurs sur la nécessité de s'enregistrer.
- Utilisation de beaucoup de papier : Le processus implique trop de travail en utilisant
- Des papiers dont le stockage pourrait être dangereux ;
- Encombrants avec la taille de la population.
- Perte des formulaires d'inscription : Quelques fois ;
- Perdre après avoir été rempli avec les détails des électeurs, dans la plupart des cas,

- Elles sont difficiles à suivre et donc beaucoup ne sont pas enregistrés, même s'ils sont des citoyens en âge de voter.
- Problème des longues files d'attentes observées devant les bureaux de vote.
- La situation géographique des différents bureaux de vote et l'embrouille quant au choix des bureaux de vote restent un frein pour certaines personnes.
- La faiblesse des taux d'inscription sur les listes électorales et de la participation citoyenne aux scrutins.
- Les pertes de temps de mobilisation du personnel humain : dépouillement, calculs et édition des résultats.
- Erreurs lors de la saisie des données : L'erreur est humaine, il est très donc peu probable pour les employés d'être à cent pour cent efficace dans la saisie des données.
- Les erreurs dues au décompte manuel des suffrages.
- Le coût des élections.
- Les élections électroniques avec de machines à voter nécessitent le déplacement de tous les participants au vote, or il est difficile de convaincre tout le monde de faire le déplacement alors qu'il serait facile de voter de chez soi, de façon mobile.
- Les avantages seraient multiples. Néanmoins, la réalisation d'un système de vote mobile n'est pas une tâche facile, car le vote mobile pose un double problème qui est celui de l'anonymat et de la confidentialité.

### **1.3. Problèmes rencontrés sur l'organisation des élections**

L'organisation d'une élection peut s'avérer très pénible. Les élections organisées en République Démocratique du Congo n'en font pas exception. Certains problèmes sont donc visibles sur l'organisation des votes et peuvent constituer des points majeurs soulevés. Les problèmes les plus marquants sont : le temps et le travail fastidieux, le taux de participation faible, la participation simultanée à plusieurs élections.

#### **1.3.1. Le temps et le travail fastidieux sur l'organisation des élections**

D'après la Commission Electorale Nationale Indépendante, il consacre beaucoup de temps à l'organisation des élections présidentielles. Le recensement des candidats ainsi que la vérification de leur éligibilité leur coûtent beaucoup de temps ; lui et son équipe doivent vérifier pour chaque candidat les critères de sélection.

Pendant la phase du vote, ils doivent faire le tour de d'un électeur à l'autre pour procéder à l'élection ; ceci a un impact en termes de ressources et du fil des temps. Par exemple pour les citoyen Congolais, il faudra que le votant se déplace jusqu'au bureau de vote et ainsi procéder au vote. Et plus encore, ils doivent attendre à des heures creuses ou bien même interrompre certaines activités afin de procéder à l'élection.

L'organisation des élections présidentielles est très pénible et nécessite beaucoup de ressources et de travail fastidieux de la part de l'administration.

De même que pour les autres élections, l'organisation peut prendre beaucoup de temps. Il arrive qu'il y ait décalage de la date du scrutin, car le nombre de candidats inscrits est faible. Et dans la phase de proclamation, le dépouillement et le comptage des voix prennent du temps



et les risques d'erreur (erreur de comptage), néanmoins faibles ne sont pas à négliger.

### **1.3.2. Problème de transparence et de fiabilité**

Le principe d'une élection est que les électeurs ainsi que les candidats acceptent de se conformer aux résultats. Il faudra donc faire confiance en la sincérité du scrutin. Le problème de transparence repose surtout sur le dépouillement et la proclamation des résultats. Le vote à l'urne semble être transparent, mais peut-on se fier aux personnes faisant le dépouillement des votes. À travers mon entretien à la CENI, il parlait de la transparence du vote, c'est-à-dire que même les votes pouvaient être truqués par ceux qui l'organisent dans la mesure où ces derniers pouvaient falsifier les résultats des votes. Ce n'est pas une affirmation de sa part, mais c'est une possibilité. Ce qui constitue un risque non-négligeable. La question de la fiabilité du vote repose donc essentiellement sur les acteurs impliqués dans l'organisation. La neutralité des superviseurs doit impérativement être prise en compte, car si elle est négligée cela peut être un risque énorme au niveau du dépouillement.

Nous avons aussi un autre problème lié au vote par procuration, où un électeur, peut décider de déléguer son vote à un mandataire pour qu'il le fasse à sa place. La confiance totale n'est pas assurée car ce même mandataire peut être tenté de modifier le choix de vote défini au départ par l'électeur mandant. Le vote par procuration nécessite une attention particulière car sa transparence est toujours remise en question.

### ***1.3.3. Le faible taux de participation***

L'absence des participants a un impact aussi sur le taux de participation.

En effet, beaucoup d'électeurs ne participe pas au vote à cause de :  
 Fil du temps, d'autres partent en voyage et par conséquent, ils ratent la

phase de scrutin. Le faible taux de participation influe considérablement sur le suffrage valablement exprimé.

#### **1.4. Problématique du sujet**

Depuis très longtemps, l'organisation des élections se faisait dans de telles conditions. Le vote électronique avec des machines à voter, malgré ces bénéfices, est très pénible dans son déroulement ; un pays comme la République Démocratique du Congo a besoin de s'acquérir de moyens plus performants et rapides favorisant pleinement la gestion des élections. Notre application essayera d'apporter des améliorations dans l'organisation et le déroulement des votes tout en garantissant un accès sécurisé.

#### **1.5. Solution proposée :**

Afin de pallier aux défaillances observées, nous proposons de mettre un système mobile pour le processus de vote, pour contribuer à mettre fin aux pratiques anciennes et perverses, en dépit des aléas et suspicions qu'on prête à ce genre d'opération. Ce système prendra en compte les fonctionnalités suivantes :

- Inscriptions des électeurs en ligne.
- Consultation de la liste des candidats et s'informer sur eux.
- Vote en ligne.

Notre application utilise des technologies sûres et fiables afin d'améliorer plusieurs aspects du déroulement des élections. Elle consiste essentiellement à conjuguer des outils informatiques et mobiles afin de faire la majorité des opérations qui sont effectuées pendant la journée des élections.

Un système de vote mobile est découpé en scénarii, ceux-ci peuvent être divisés en plusieurs phases :

1. La phase d'enregistrement : électorale « pour toutes les personnes éligibles », ceci après enregistrement des votants éligibles pour le vote et la publication de celle.
2. La phase de vote : phase au cours de laquelle les votants envoient leur bulletin de vote en utilisant les facilités de communication offertes par le réseau ;
3. La phase de décompte : bulletins de vote, le déclenchement du décompte des votes, la publication des résultats sur le réseau et la mise à la disposition de ceux.

### **1.6. Objectifs spécifiques**

Les objectifs de notre application sont de :

- Réduire le temps de travail sur l'organisation, la supervision du processus de vote et le dépouillement des voix (qui se fera automatiquement) ;
- Gérer la fiabilité et la transparence du vote en assurant la vérification et la confidentialité du vote ;
- Gérer simultanément plusieurs élections ouvertes au niveau de la plateforme ;
- Supprimer le vote par correspondance en permettant la mise en place d'un système décentralisé favorisant le vote pour toutes les personnes concernées ;
- Favoriser la sécurité du vote tout en gérant le contrôle d'accès à la plateforme selon le profil d'utilisateur ;
- Permettre la publication des listes imprimées des candidats retenus et des résultats des scrutins de vote.

### ***1.6.1. Les enjeux sur l'utilisation des systèmes de vote par internet***

La grande question qui subsiste toujours sur l'utilisation des systèmes de vote en ligne est l'assurance totale de la sécurité et de la transparence de vote. L'utilisation de ces types de système reste toujours critiquée par des experts qui jugent que les risques d'attaque sont énormes. Il y a aussi l'utilisation qui peut s'avérer très délicate. Certains de ces systèmes nécessitent une petite formation à un mois des échéances électorales. Ce qui peut s'avérer très pénible en termes de coût et de ressources.<sup>1</sup>

Malgré toutes ces limites, nous considérons qu'ils sont efficaces dans la mesure où le processus de vote devient rapide et efficace. On perd moins de temps dans l'organisation et le comptage des voix et le délai de proclamation des résultats est court. Les risques d'erreur de comptage des voix sont inexistantes. Pour renforcer les moyens de sécurité, on peut faire appel à des techniques de cryptographie modernes très puissantes ou même utiliser des serveurs proxy hautement efficace pour garantir la sécurité. Pour des élections à faibles enjeux, les risques d'attaques sont minimes.

### **1.7. Etude des besoins**

Dans cette section du chapitre, nous nous intéressons aux besoins de vote mobile à travers les spécifications fonctionnelles et non fonctionnelles pour aboutir à une application de qualité selon les besoins des utilisateurs.

---

<sup>1</sup> <https://www.Commençamarche.com>

### **1.7.1. Besoins fonctionnels :**

Les besoins fonctionnels représentent les principales fonctionnalités du système. Ces Besoins proviennent généralement des utilisateurs du système. Les fonctionnalités à Développer dans notre application comporte (2) principaux modules à savoir :

- Le module utilisateur (Electeur, Candidat).
- Le module de vote.
- Le Module de Gestion :(Administrateur)

### **1.7.2. SPECIFICATION DES BESOINS**

Dans cette partie, il est nécessaire de délimiter le contexte du sujet en faisant une étude fonctionnelle du futur système. Il s'agit donc de montrer l'ensemble des acteurs agissant au sein de notre système et d'identifier les cas d'utilisation spécifiques.

#### **A. Identification des acteurs**

Un acteur est l'archétype de l'utilisateur (personne, processus externe, ...) qui interagit avec le système. Par défaut, c'est un acteur principal, c'est-à-dire qu'il agit directement sur le système et en attend des résultats ou bien, il peut être un acteur secondaire qui est souvent sollicité pour des informations supplémentaires.

Dans notre cas, le tableau ci-dessous détaille l'ensemble de nos acteurs agissant sur le système.

**Tableau 1: Identification des acteurs**

ACTEURS	DESCRIPTIONS	ROLES
<b>Candidat</b>	Pour chaque type d'élections, on a un type spécifique de candidat	Son rôle au système est pour voter
<b>Electeur</b>	Les électeurs doit être un citoyen congolais,	Son rôle au système est pour voter
<b>Administrateur</b>	Administrateur est à la gestion de plateforme	Son rôle est de gérer tous les utilisateurs ainsi que les décomptages

### **B. Identification des fonctionnalités**

Les fonctionnalités sont les actions ou services pouvant être rendus par le système en cas de sollicitation de l'acteur concerné. Ces services peuvent être externes (nécessitant un acteur) ou bien, interne, c'est-à-dire non-visible, mais fonctionnant en interne s'il y a le démarrage d'un processus donné. Dans notre étude, nous avons pu lister un ensemble de fonctionnalités pouvant mener à bien l'organisation et le processus de vote dans notre système. Le tableau ci-dessous montre l'ensemble des fonctionnalités.

Un protocole de vote, pour être utilisable doit vérifier un certain nombre de propriétés, nous présentons brièvement chacune d'entre elles.

Ces propriétés sont :

1. Secret des votes : personne ne doit pouvoir faire le rapprochement entre un électeur et son vote.

2. Éligibilité : seuls les électeurs enregistrés doivent pouvoir voter. Cette propriété est vérifiée si l'intrus ne peut pas obtenir les paramètres d'authentification (Numéro National) de la phase d'enregistrement lui permettant de continuer le processus.
3. Pas de double vote : aucun électeur ne doit pouvoir voter deux fois lors d'une même élection. Toutefois, dans le cas contraire son vote ne doit pas être comptabilisé plus d'une fois. Il ne faudrait pas rejeter les votes valides.
4. Pas de résultat Partiel (Équité) : personne ne doit être capable d'obtenir des résultats partiels car la connaissance de ces résultats pourrait influencer les électeurs n'ayant pas encore voté.
5. Précision : l'élection est précise si le vote n'est pas altéré, par conséquent les résultats du vote ne doivent pas être modifiés en ajoutant des votes invalides ou en changeant le contenu des bulletins par exemple (intégrité).
6. Complétude : Un vote valide doit être comptabilisé et un vote invalide ne doit pas être comptabilisé.
7. Receipt-freeness (Sans Reçu) : elle représente une forme forte de confidentialité. Définie par Benaloh en 1994 comme suit : aucun électeur ne doit être capable de prouver la manière dont il a voté, d'obtenir ou d'être capable de construire un reçu de son vote, (c'est à dire un document prouvant comment il a voté). Cette propriété est importante pour la prévention contre la vente des votes.<sup>2</sup>
8. Résistance à la coercition : la coercition consiste à forcer quelqu'un à voter d'une certaine manière et de s'en assurer par la suite. Les propriétés ci-dessus sont hiérarchisées en deux niveaux. Un niveau inférieur constitué des exigences de base (éligibilité, secret, précision,

---

<sup>2</sup> Ibidem

vérifiabilité individuelle, ...) devant être accomplies dans chaque système de vote. Un niveau supérieur (vérifiabilité universelle, Receipt-freeness, résistance à la coercition) regroupe les exigences avancées pour les systèmes de vote.<sup>3</sup>

---

<sup>3</sup> Ibidem



### 1.7.3. Tableau d'ordonnancement des tâches

POSTE	TACHE	COMMENTAIRE
<b>100 Utilisateur</b>	101	- Présentation et authentification d'utilisateur au système
	102	- Choisir un candidat parmi les candidats affichés sur la liste.
<b>200 Formulaire de vote</b>	201	- Le système envoie une requête à la BDD pour vérifier que l'utilisateur est enregistré à la base des données.
	202	- Si oui ! le système envoie un message « Vote effectuer »
<b>300 Base de données</b>	301	- BDD Renvoi au système affiche un msg « Connexion réussie » si la vérification se termine par succès ensuite l'interface d'accueil s'affiche
	302	- Vérifier dans la BDD que l'utilisateur n'a pas encore voté.

#### **1.7.4. Besoins non fonctionnels :**

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur, ce qui fait qu'ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

- Fiabilité : l'application doit fonctionner de façon cohérente sans erreurs ;
- Les erreurs : l'application doit les signaler par des messages d'erreurs ;
- Ergonomie : l'application doit être adaptée à l'utilisateur sans qu'il fournisse trop d'effort (utilisation claire et facile) ;
- Efficacité : l'application doit permettre l'accomplissement de la tâche avec le minimum de manipulations.
- Sécurité : l'application doit être sécurisée au niveau des données : authentification et contrôle d'accès.

#### **1.8. Cadre du projet :**

Dans le cadre de notre projet de fin d'étude au sein de la Commission Electorale Nationale Indépendante nous avons eu comme tâche de concevoir et développer une application mobile pour le vote des élections présidentielles. L'objectif visé est de mettre en place une stratégie et une méthodologie cohérente afin d'accroître la confiance des populations dans le processus électoral et d'obtenir la mobilisation et la participation effective des électeurs potentiels à la vie politique du pays, basée sur un plan à court, moyen et long terme, s'avèrent nécessaires. Il s'agit donc de dépasser l'approche ponctuelle pour adopter une approche s'inscrivant sur la durée afin d'atteindre des résultats tangibles.

**Conclusion :**

Dans ce chapitre nous avons présenté une étude de l'existant du vote, les lacunes qu'il comprend ainsi que les solutions que nous proposons pour pallier ces problèmes, nous avons aussi cité les besoins fonctionnels et non fonctionnels qui sont indispensables pour mieux faciliter le travail à réaliser. Dans le chapitre suivant nous allons aborder l'étude conceptuelle de notre application mobile, tout en mentionnant quelques scénarios possibles, les acteurs et les diagrammes.

## CHAPITRE II ETAPE CONCEPTUELLES

### II.1. Introduction :

La réalisation d'une application doit être impérativement précédée d'une méthodologie d'analyse et de conception qui a pour objectif de permettre de formaliser les étapes préliminaires du développement afin de rendre ce développement plus fidèle aux besoins d'utilisateur.

La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités et la phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation. La phase de modélisation (analyse et conception) est la plus cruciale du processus du développement d'un logiciel.

### Section 1 : Présentation de langage UML :

#### A. Introduction :

UML (Unified Modeling Language, que l'on peut traduire par "langage de modélisation unifié) est une notation permettant de modéliser un problème de façon standard.<sup>4</sup> Ce langage est né de la fusion de plusieurs méthodes existantes auparavant, et est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

#### B. Définition :

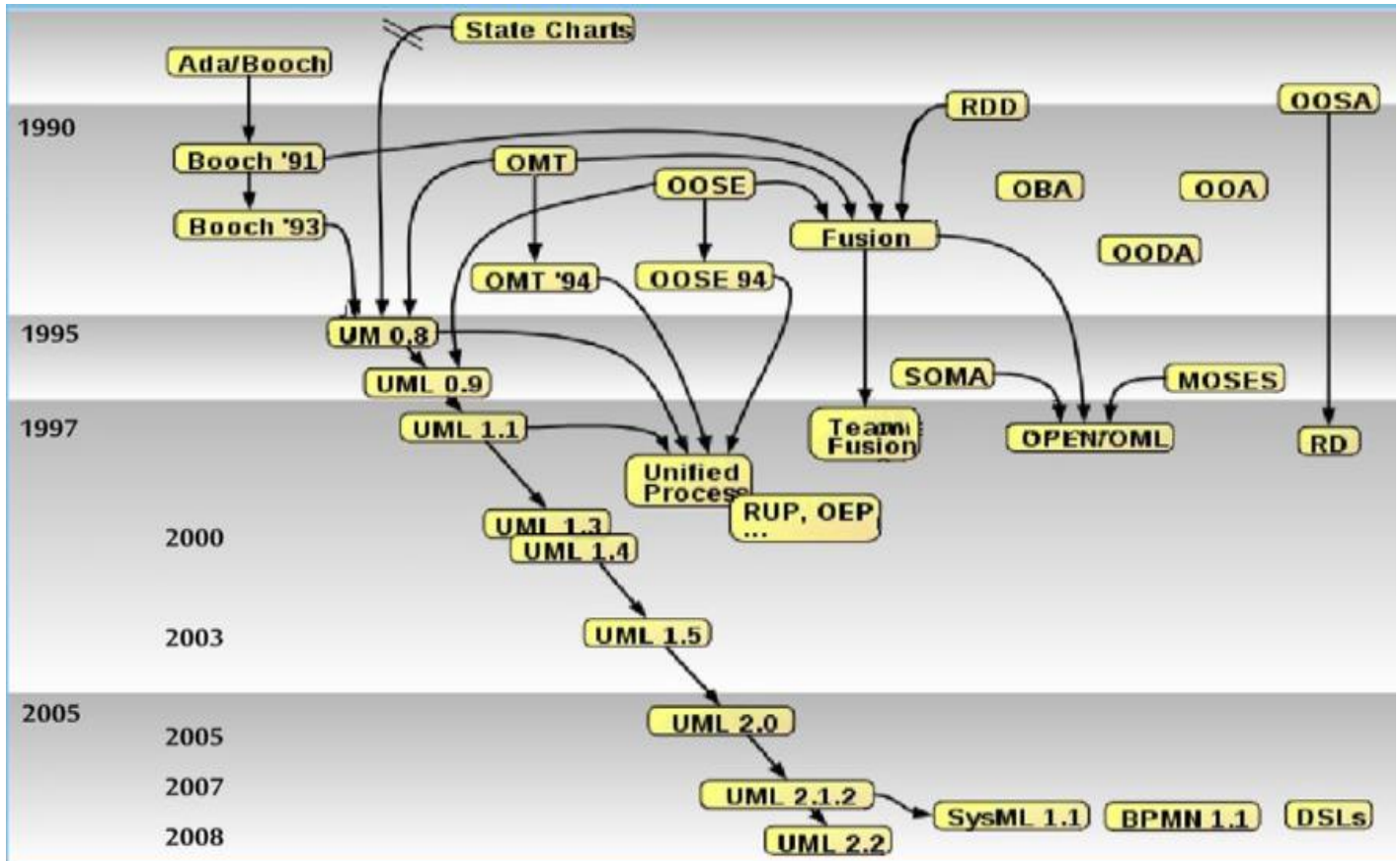
UML est un langage de modélisation graphique et textuel destiné à comprendre et à décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. En effet UML est un langage avec une syntaxe et des règles bien définies qui tentent à réaliser les buts décrits

---

<sup>4</sup> Note de cours L2 Question Spéciale de Conception de Système d'Information , op.cit

grâce à une représentation graphique formée de diagrammes et une modélisation textuelle qui vient enrichir la représentation graphique.<sup>5</sup>

### C. Evolution d'UML<sup>6</sup>



## 2.2. PRESENTATION DES DIAGRAMME

### 2.2.1. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est une représentation des principales fonctionnalités nécessaires à l'étude de notre système. Il décrit le système étudié en privilégiant le point de vue utilisateur.

Il permet ainsi de recueillir, d'analyser et d'organiser les besoins, et de recenser les principales fonctionnalités d'un système. Il montre aussi les interactions existant entre acteurs et cas d'utilisation, entre cas d'utilisation (inclusion, extension, généralisation), et entre acteurs (généralisation).

<sup>5</sup> <https://www.Wikipedia.com>

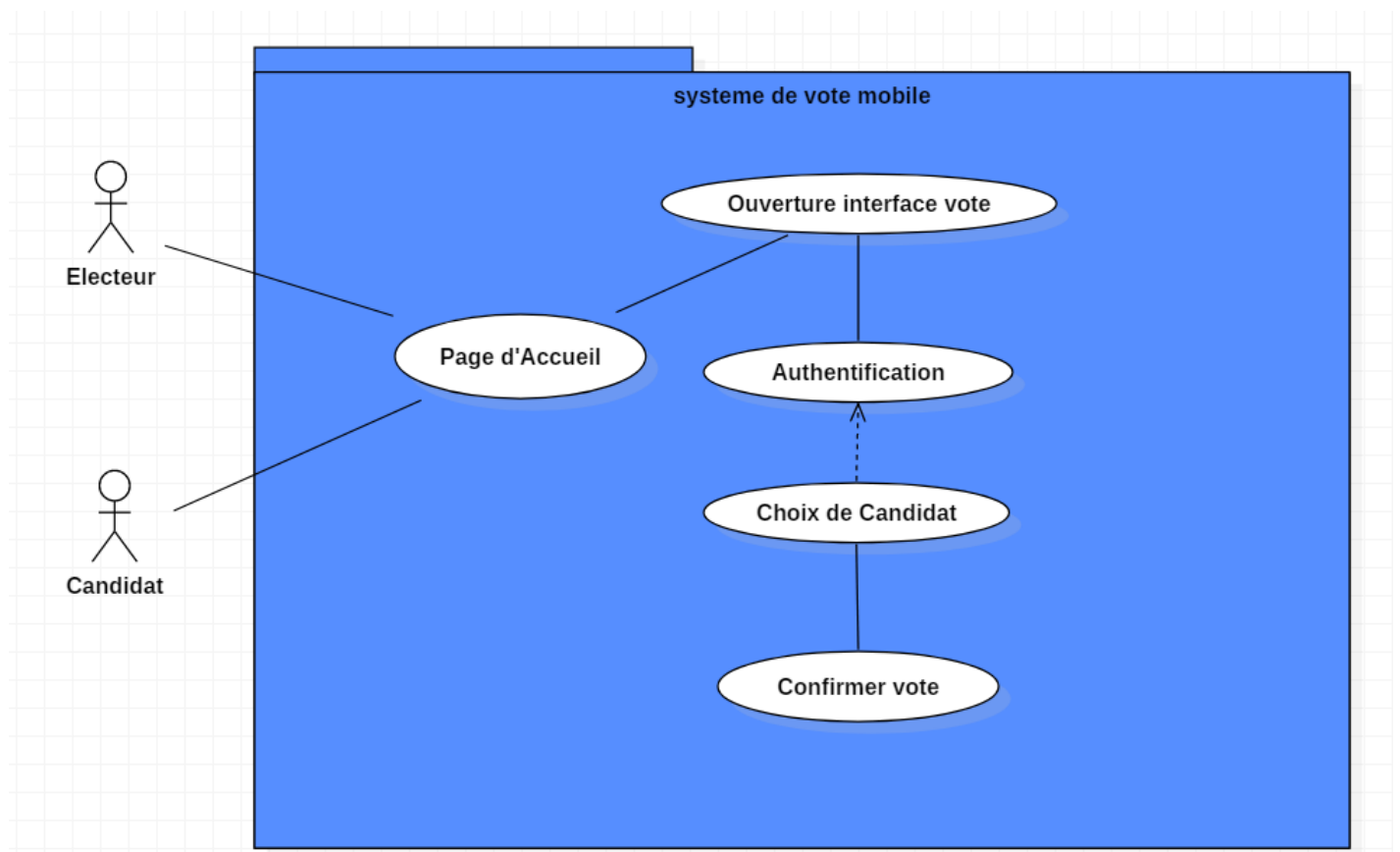
<sup>6</sup> <https://prezi.com/p/2ikdh0zd8x4f/historique-de-uml/>

Dans notre cas, nous avons trois diagrammes de cas d'utilisation selon les acteurs concernés.

Dans le diagramme, ci-dessus, nous voyons que l'administrateur a un grand nombre de fonctionnalités à gérer ; et pour chaque fonctionnalité, il a besoin de s'authentifier. S'il y a de nouveaux membres, il doit les enregistrer au niveau de la base de données de l'application. Il crée et attribue une élection à un superviseur.

### a. Diagramme de cas d'utilisation des électeurs et candidats

Le diagramme ci-dessous montre quelques fonctionnalités faites par les électeurs et les candidats :



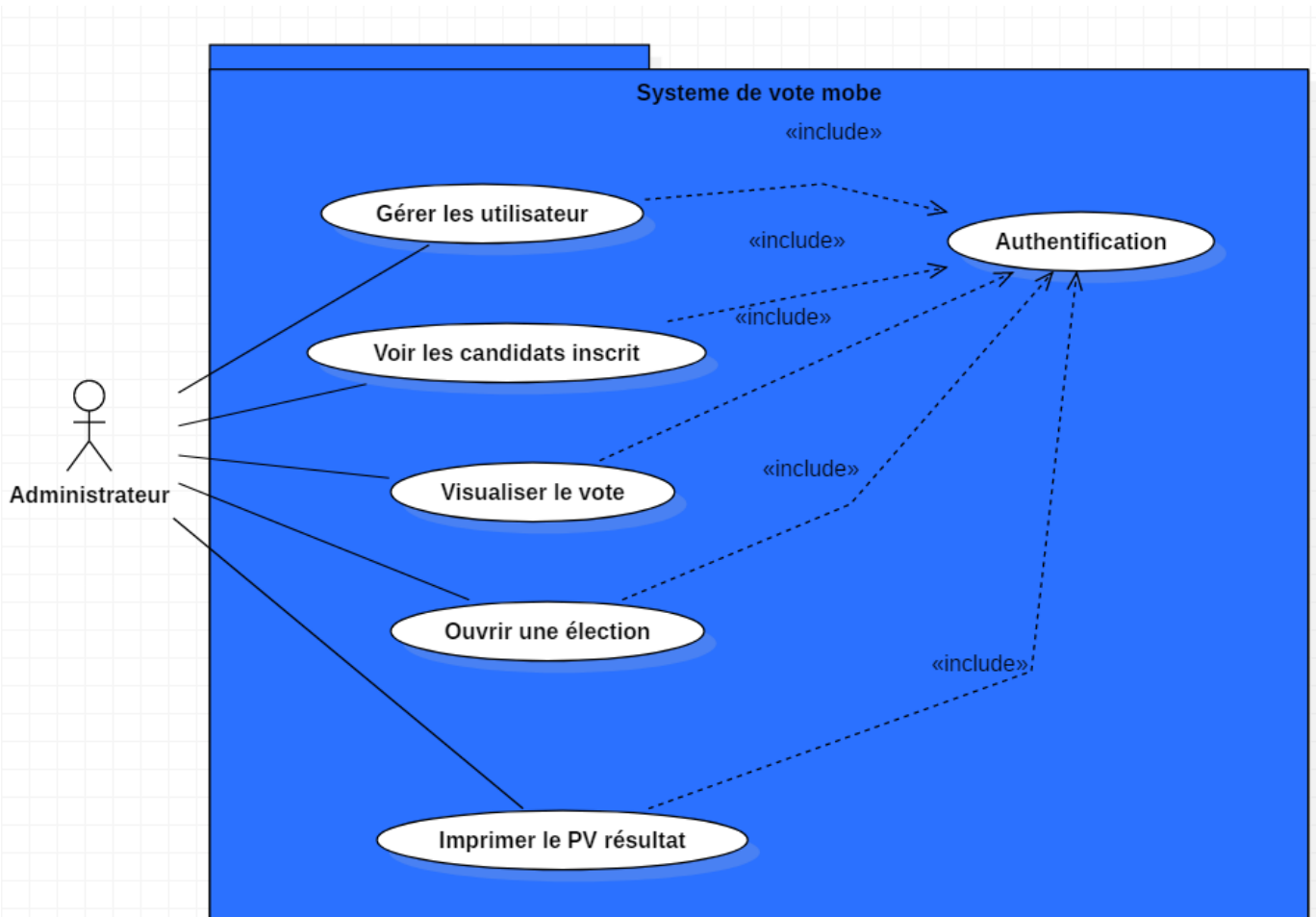
Dans ce diagramme, nous voyons que l'électeur peut déposer sa candidature. Cette fonctionnalité est activée lorsqu'il n'y a pas encore

déroulement du scrutin. Tous les utilisateurs du système sont d'abord considérés comme des électeurs. L'électeur et le candidat peuvent voter.

Le candidat quant à lui peut à chaque fois modifier ses informations ou même retirer sa candidature.

### **b. Diagramme de cas d'utilisation de l'administrateur**

Le diagramme de cas d'utilisation de l'administrateur permet de représenter les fonctionnalités auxquelles cet acteur a accès à notre système.



Dans le diagramme, ci-dessus, nous voyons que l'administrateur a un grand nombre de fonctionnalités à gérer ; et pour chaque fonctionnalité, il a besoin de s'authentifier. S'il y a de nouveaux membres, il doit les enregistrer

au niveau de la base de données de l'application. Il crée et attribue une élection à un superviseur.

### 3. Description des cas d'utilisation

Dans cette partie, nous allons faire la description de quelques cas d'utilisation de notre diagramme. La description des cas se fera sous forme de tableau.

#### a. Description du cas d'utilisation « s'authentifier » Général

**Tableau 1 : Description du cas d'utilisation "s'authentifier"**

IDENTIFICATION	
<b>Cas</b>	1
<b>Nom</b>	S'authentifier
<b>Acteur(s)</b>	tous les utilisateurs du système
<b>Description</b>	Avant toute manipulation du système, faudra procéder à une authentification
<b>Auteur</b>	KONGA MASSAMBA Théophile
<b>Date</b>	25/12/2023
<b>Pré conditions</b>	Il faudra s'authentifier
<b>Démarrage</b>	l'utilisateur lance l'application et essaie d'entrer dans une fonctionnalité
SCENARIO NOMINAL	
<b>1</b>	Au lancement de l'application, un formulaire d'authentification est lancé
<b>2</b>	Dans ce formulaire, il demande à l'utilisateur de donner son login et son mot de passe
<b>3</b>	Si les informations saisies par l'utilisateur sont correctes, il lui envoie un message confirmant



	connexion réussie et on note une ouverture du formulaire voulu
SCENARIOS ALTERNATIFS	
<b>1a</b>	L'utilisateur peut décider de quitter même l'authentification
<b>2a</b>	L'utilisateur peut entrer un login et un mot de passe incorrect où il devra réessayer la saisie
<b>3a</b>	L'accès sera refusé à toute personne ne faisant pas partie de la base (qui n'est ni un électeur, ni un candidat, ni un administrateur)

**b. Description du cas d'utilisation « Paramétrer élection »**

**Tableau 2 : Description du cas d'utilisation "paramétrer élection"**

IDENTIFICATION	
<b>Cas</b>	2
<b>Nom</b>	Paramétrer une élection
<b>Acteur(s)</b>	Administrateur
<b>Description</b>	Il permet l'ajout ou bien l'ouverture d'une élection au niveau de la plateforme et qui sera pris en compte pour l'ouverture du scrutin
<b>Auteur</b>	KONGA MASSAMBA Théophile
<b>Date</b>	25/12/2023
<b>Pré conditions</b>	Il faudra s'authentifier
<b>Démarrage</b>	L'utilisateur sera redirigé vers le menu administrateur Aller vers l'option paramétrer élection
SCENARIO NOMINAL	
<b>1</b>	L'authentification réussie, l'utilisateur sera redirigé

	vers le menu Administrateur
<b>2</b>	Il peut donc aller vers l'option paramétrer élection
<b>SCENARIOS ALTERNATIFS</b>	
<b>1a</b>	En cas d'authentification invalide, l'utilisateur n'accèdera pas au système
<b>2a</b>	Il peut décider de quitter le menu en se déconnectant
<b>3a</b>	Il peut quitter la page de paramétrage d'une élection et se déconnecter

**c. Description du cas d'utilisation « s'authentifier »**

**Tableau 3 : Description du cas d'utilisation "s'authentifier"**

<b>IDENTIFICATION</b>	
<b>Cas</b>	3
<b>Nom</b>	S'authentifier
<b>Acteur(s)</b>	Electeur, Candidat
<b>Description</b>	<b>Electeur</b> : 1. doit être déjà inscrit dans l'application 2. N'as pas encore voté <b>Candidat</b> : Même contrainte d'électeur
<b>Auteur</b>	KONGA MASSAMBA Théophile
<b>Date</b>	25/12/2023
<b>Pré conditions</b>	Il faudra s'authentifier
<b>SCENARIO NOMINAL</b>	
<b>1</b>	<ul style="list-style-type: none"> <li>➤ Taper votre nom</li> <li>➤ Numéro de la carte d'électeur</li> <li>➤ Adresse Gmail</li> <li>➤ Choisir un candidat parmi les candidats sur la liste</li> </ul>

2	➤ Appuyer sur « Confirmer ».
3	➤ Le système envoie une requête à la BDD pour vérifier la validité des identifiants et la 2 <sup>ème</sup> Contrainte, et affiche un msg « Connexion réussie » si la vérification se termine par succès ensuite un message s'affiche « vote réussie »
<b>SCENARIOS ALTERNATIFS</b>	
1a	➤ Si les identifiants sont incorrects ou incomplets, le système affiche un message d'erreur.
2a	Si l'électeur a déjà utilisé l'application et a choisit un candidat un msg s'affiche « Tu as déjà voté(e) ».

**d. Description du cas d'utilisation « S'inscrire »**

**Tableau 3 : Description du cas d'utilisation "S'inscrire"**

<b>IDENTIFICATION</b>	
<b>Cas</b>	4
<b>Nom</b>	S'inscrire
<b>Acteur(s)</b>	Candidat, Electeur
<b>Description</b>	<b>Electeur</b> : doit figurer dans la BDD des personnes ayant une carte d'électeur. <b>Candidat</b> : même contrainte électeur.
<b>Auteur</b>	KONGA MASSAMBA Théophile
<b>Date</b>	25/12/2023
<b>Pré conditions</b>	Il faudra s'inscrire
<b>SCENARIO NOMINAL</b>	
1	➤ Saisir votre nom, le numéro de la carte d'électeur et l'adresse Gmail
2	➤ Appuyer sur enregistrer.

3	➤ Le système envoie une requête à la BDD pour vérifier que l'utilisateur a déjà une carte ou non.
<b>SCENARIOS ALTERNATIFS</b>	
1a	➤ Si le numéro de la carte entré n'existe pas ou bien un nom erroné. Le système affiche un msg « Tu n'as pas une carte d'électeur »
2a	➤ Si une erreur se produira lors de la sauvegarde du nom d'utilisateur et le numéro de la carte d'électeur, l'électeur sera informé.

e. Description du cas d'utilisation « Voter »

Tableau 4: Description du cas d'utilisation "Voter"

<b>IDENTIFICATION</b>	
<b>Cas</b>	5
<b>Nom</b>	Voter
<b>Acteur(s)</b>	Electeur, Candidat
<b>Description</b>	<b>Electeur</b> : avoir fait l'authentification <b>Candidat</b> : même contrainte d'électeur
<b>Auteur</b>	KONGA MASSAMBA Théophile
<b>Date</b>	25/12/2023
<b>SCENARIO NOMINAL</b>	
1	➤ Choisir un candidat parmi les candidats affichés sur la liste.
2	➤ Appuyer sur « Voter »
3	➤ Vérifier dans la BDD que l'utilisateur n'a pas encore voté.
5	➤ Appuyer sur « Confirmer » et le choix va être

	enregistré dans la BDD.
<b>6</b>	➤ Le système affiche un msg « Vote réussit ».
<b>SCENARIOS ALTERNATIFS</b>	
<b>1a</b>	➤ Si le candidat ne confirme pas son choix, il vas choisir un autre candidat de la liste

### **3.1. Analyse des besoins fonctionnels du système**

Dans cette partie, nous allons faire l'analyse des besoins fonctionnels du système. Chaque cas d'utilisation présente tout un ensemble de processus dans son fonctionnement. Nous choisirons l'analyse de trois cas d'utilisation en présentant leurs diagrammes d'activité et de séquence.

### **3.2. Etude des besoins fonctionnels du système**

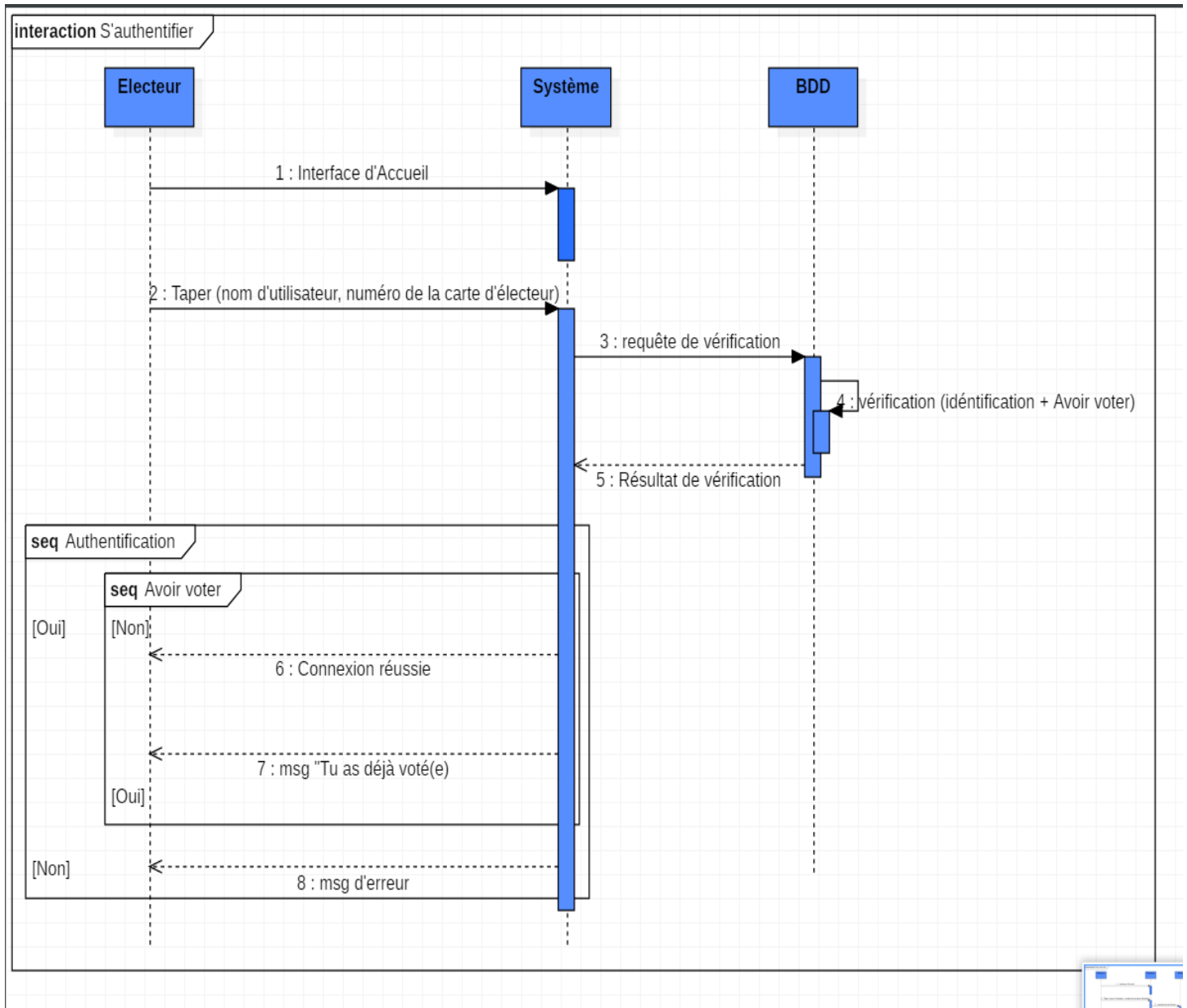
Dans cette partie, nous allons faire l'analyse des besoins fonctionnels du système. Chaque cas d'utilisation présente tout un ensemble de processus dans son fonctionnement. Nous choisirons l'analyse de trois cas d'utilisation en présentant leurs diagrammes d'activité et de séquence.

### **3.3. Représentation du diagramme de séquence**

#### **a. Représentation du diagramme de séquence : cas « S'authentifier »**

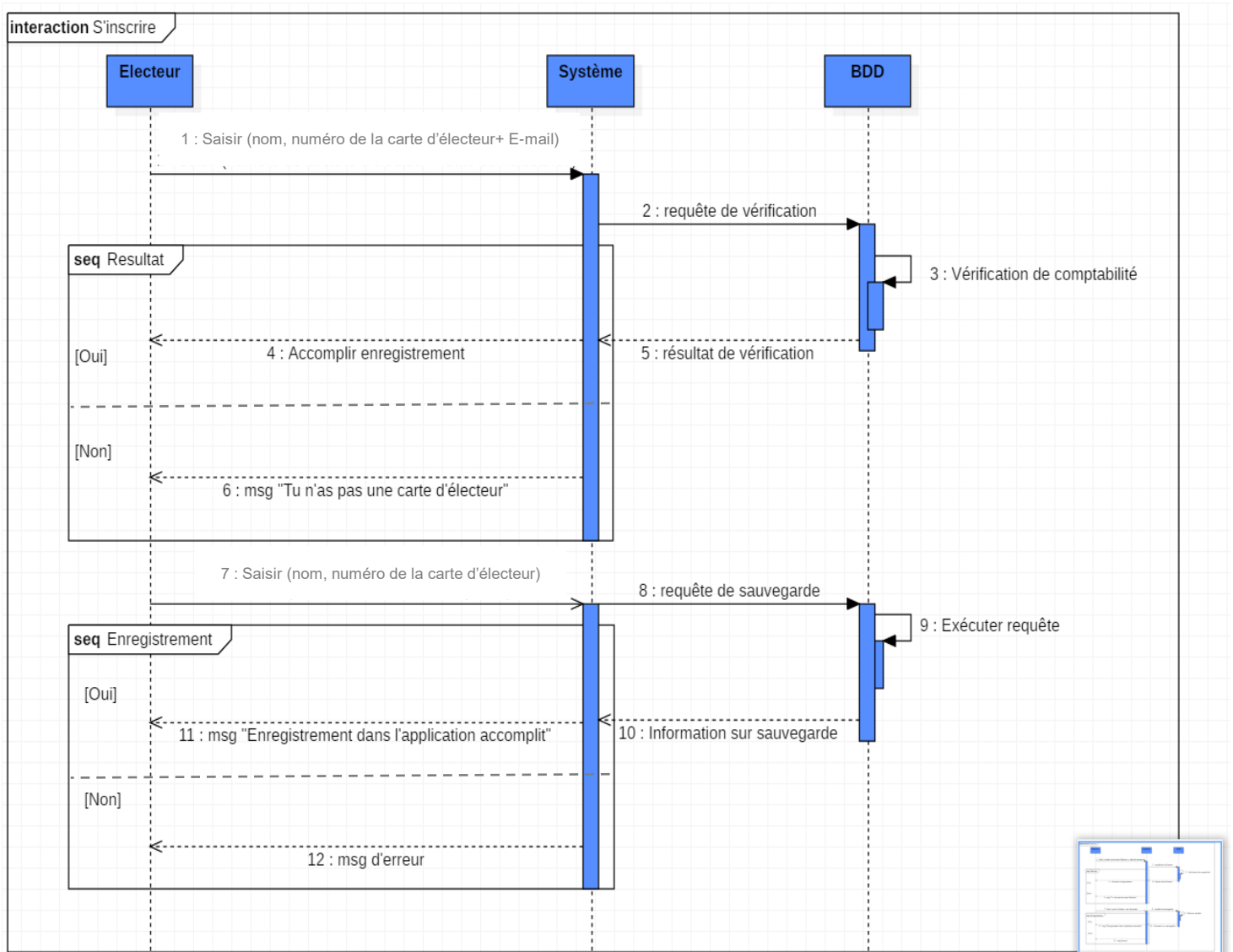
L'authentification consiste à assurer la confidentialité des données, elle se base sur la vérification des informations associées à un utilisateur (généralement un nom d'utilisateur et un mot de passe). Ces informations sont préétablies dans une base des données. Dans notre système lors d'une authentification, trois cas se présentent : les informations introduites par l'utilisateur sont incomplètes ou incorrects, dans ce cas un message d'erreur s'affiche, ou l'utilisateur à déjà voté donc le système va lui affiché un message pour l'informer qu'il ne peut plus accéder à l'application, ou bien le 3<sup>ème</sup> cas les informations saisies sont complètes, le système envoie une requête de vérification à la BDD pour déterminer leurs validités est que notre utilisateur

n'a pas encore voté, si tout va bien un message de connexion et l'interface correspondante s'affichent.



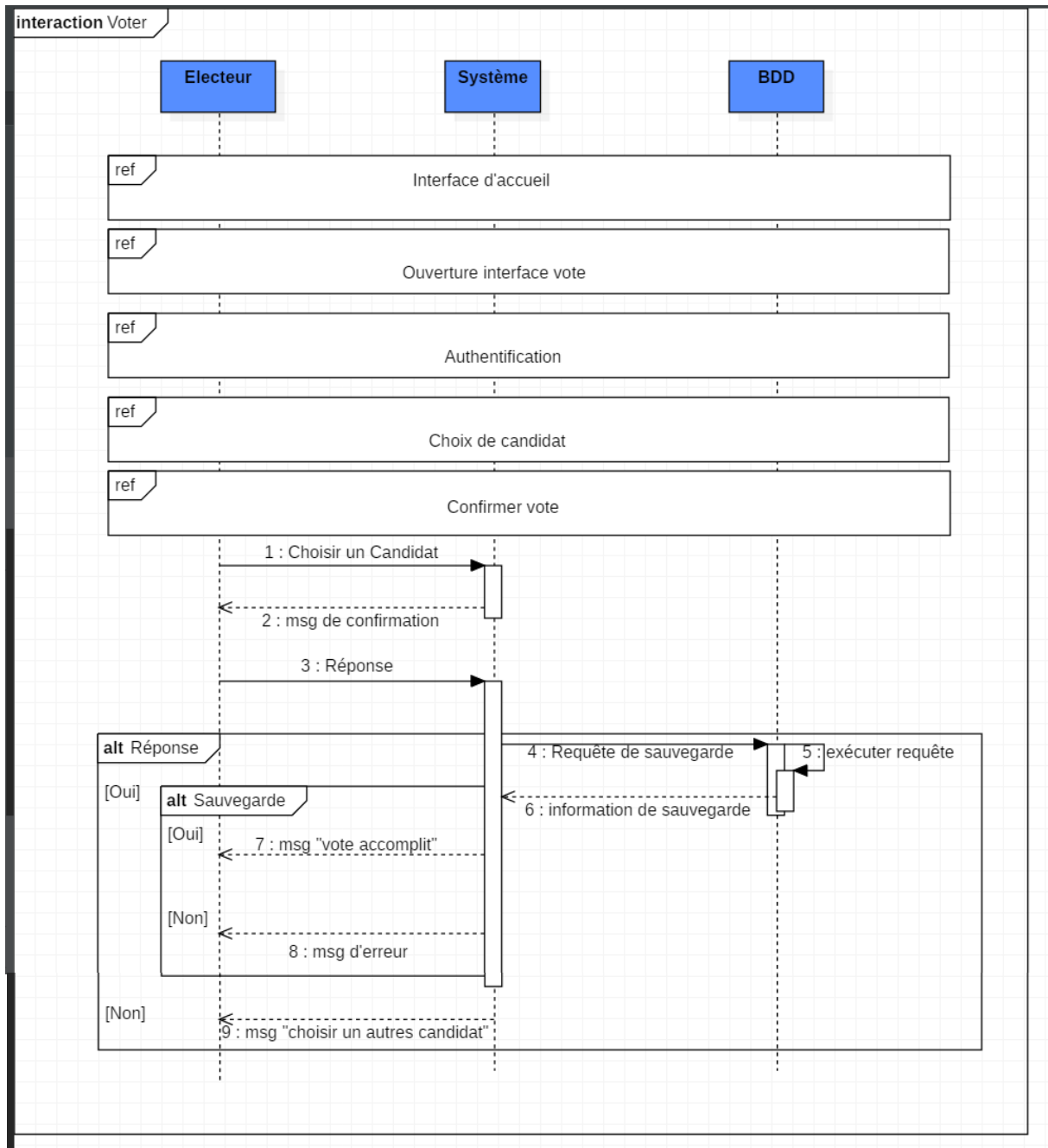
## b. Représentation du diagramme de séquence : cas « S'inscrire »

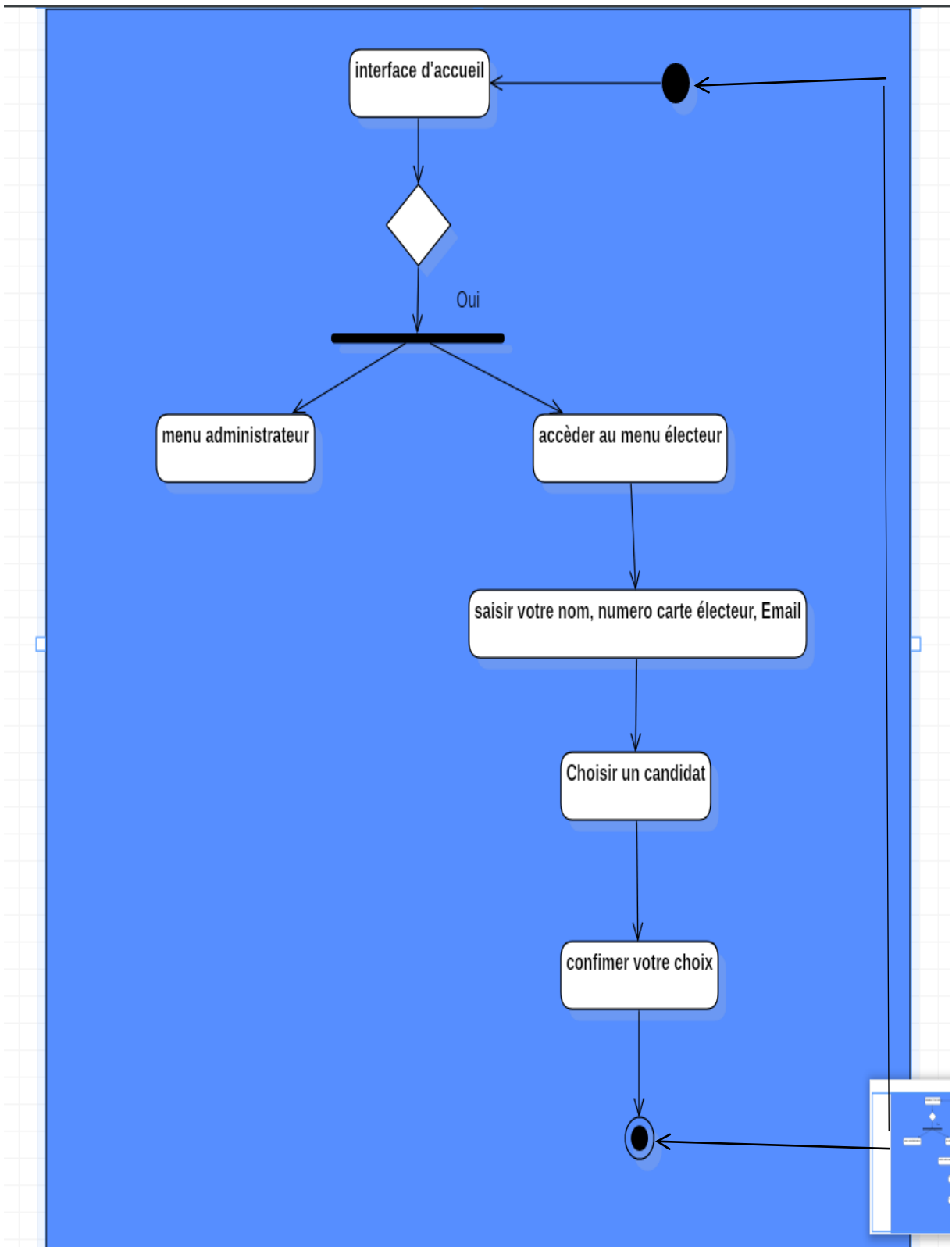
Afin de pouvoir voter, il faut s'inscrire dans l'application mais sauf les personnes qui ont déjà une carte d'électeur qui peuvent le faire ((disant que notre BDD est celle de mairie et on a ajouté tous ce qui est nécessaire pour le fonctionnement de notre application)). L'utilisateur remplit les champs numéro de la carte d'identité, date de naissance, nom d'utilisateur et un mot de passe ensuite le système vérifie si le N.N existe dans la BDD et que l'utilisateur à entrer la bonne, D.N si c'est le cas l'électeur va accomplir l'enregistrement sinon le système va informer l'utilisateur qu'il n'a pas une carte d'électeur.



L'utilisateur s'authentifie en saisissant donc son login et son mot de passe d'accès à la plateforme. Si les informations sont correctes, il sera redirigé vers le menu électeur avec l'option d'effectuer le vote. Il est nécessaire de préciser qu'effectuer vote. Une vérification interne est faite et il sera redirigé vers la liste des candidats où il pourra faire son choix.







### 3.4. Description des classes

Classe	Code propriété	Propriété	Nature	Taille	Identifiant
<b>Administrateur</b>	UserNam	UserNam	AN	20	
	Password	Password	AN	8	
<b>Utilisateur (Electeur, Candidat)</b>	Id.Utili	Id.Utili	AN	5	#
	Nom	Nom	AN	20	
	Prénom	Prénom	AN	20	
	NumCartElec	NumCartElec	AN	10	
	E-mail	E-mail	AN	30	
	Datenais	Datenais	AN	10	
	LieuNais	LieuNais	AN	50	
	Genre	Genre	AN	5	
<b>Candidat</b>	Id.Cand	Id.Cand	AN	100	#
	InfoCand	InfoCand	AN		
<b>Parti</b>	MatrPart	MatrPart	AN	5	#
	NomPar	NomPar	AN	20	
	DescrPar	DescrPar	AN	100	

#### 3.4.1. Présentation des règles de gestion

Les règles de gestion expriment et présentent les contraintes qui doivent être respectées par le modèle représentant les lois de l'univers réel modélisé dans le système d'information.

##### 3.4.1.1. Règles d'action (RA)

Ce sont l'ensemble des phrases dont les sujets et le complément sont des objets et les verbes sont des relations permettant à règle les informations entre les postes.

Voici les différentes règles d'action relative à notre gestion :

**RG 1.** - un administrateur peut gérer un ou plusieurs utilisateurs

- Un ou plusieurs utilisateurs sont gérés par un administrateur

**RG 2.** - Un candidat est voté par un ou plusieurs électeurs

- Un ou plusieurs électeurs peuvent voter un et un seul candidat

**RG2.** - Un parti doit appartenir à un ou plusieurs candidats

- Un ou plusieurs candidats doit appartenir à un parti

### 3.4.2. Recensement et description des relations

#### a. Recensement des relations

- Gérer
- Voter
- Appartenir

#### b. Description des relations

N°	Relation	Propriétaire	Dimension	Collection	Type Relation
1	<b>Gérer</b>	-	3 ou tenaire	Administrateur et Utilisateur	Père fils
2	<b>Voter</b>	-	3 ou Ternaire	Electeur et Candidat	Père fils
3	<b>Appartenir</b>	-	3 ou Ternaire	Candidat et Parti	Père et fils

### 3.5. Diagramme de Classe

Il s'agit ici d'une vue statique car on ne tient pas compte du facteur temporel dans le comportement du système, le Diagramme de classe ci-dessous modélise les concepts du système de vote ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation de ce système Notre diagramme de classe permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

## Une classe est composée d'un nom, d'attributs et d'opérations.

- Le nom d'une classe commence par une majuscule
- Le nom d'une propriété commence par une minuscule
- Les types de base (int, long, float, double, boolean) sont en minuscules
- Il n'a pas d'espace dans les noms de classes ou de propriétés
- Pour les noms composés, on fait commencer chaque mot par une majuscule

## Concepts et instance

Une instance est la concrétisation d'un concept abstrait.

Instanciation d'une classe :

- Une *classe* est un concept en cela qu'elle n'est que la spécification d'un type
- Un *objet* est une instance avec ses propres attributs et son propre état

## Encapsulation<sup>7</sup>

L'encapsulation est un principe de conception consistant à protéger le cœur d'un système des accès intempestifs venant de l'extérieur.

En UML, les modificateurs d'accès permettent de définir la visibilité des propriétés :

- Un attribut *privé* (-) limite la visibilité d'une propriété à la classe elle-même
- Un attribut *public* (+) ne limite pas la visibilité d'une propriété

---

<sup>7</sup> <https://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours01.html>

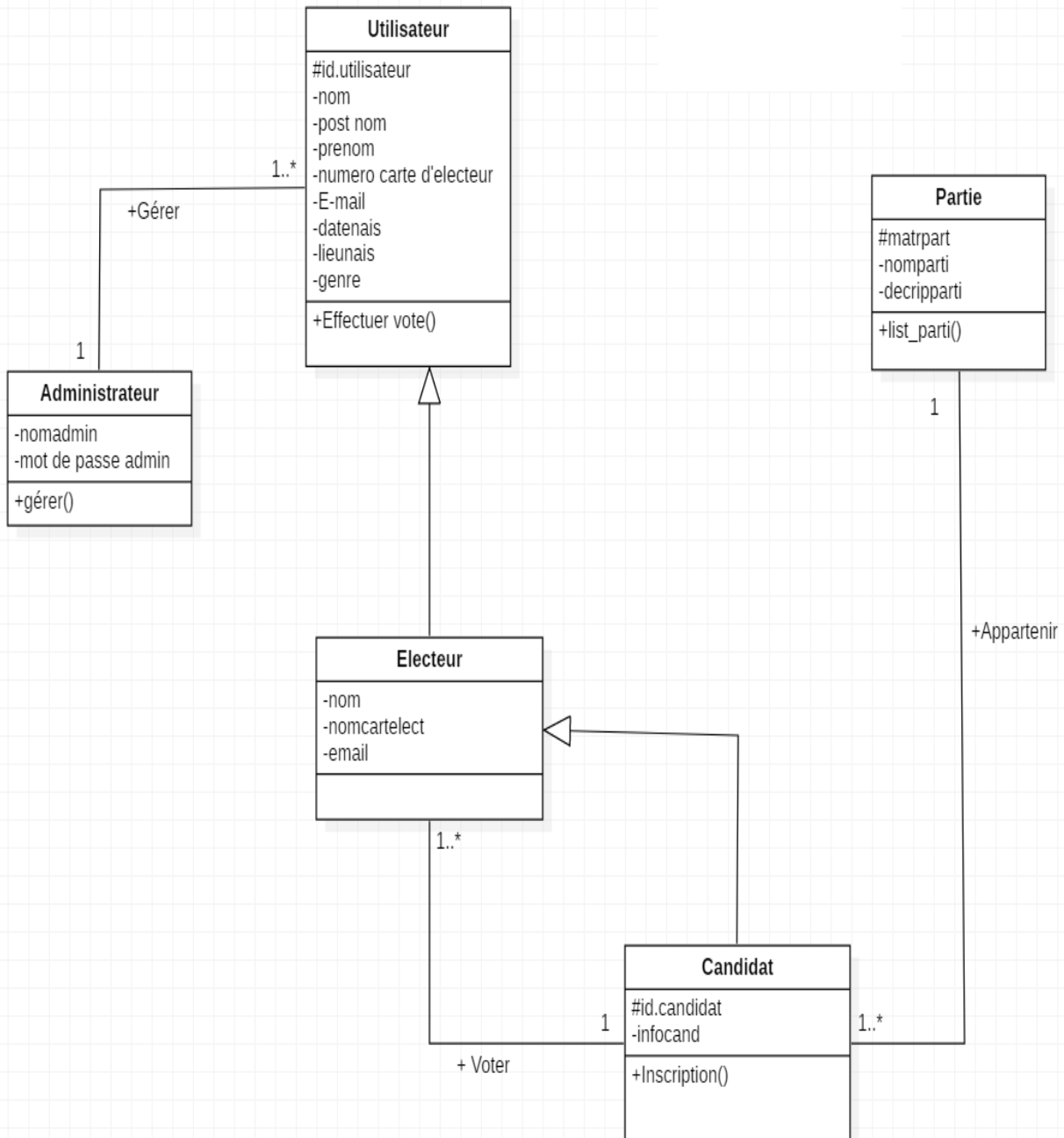
- Un attribut *protégé* (#) limite la visibilité d'une propriété à la classe elle-même et à ses sous-classes
- Le symbole (~) permet de limiter la visibilité d'une propriété au package de la classe

### 3.5.1. Présentation de Diagramme de Classe

#### ➤ Règle de Passage

- Transformation des classes : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé.
- Transformation de l'association : Nous distinguons trois familles d'associations :
  - Association 1..\* : Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.
  - Association \*.\* et n-aire et classe-association : la classe\*association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des classes connectées à l'association.
  - Association 1..1 : il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association. Si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux classes en une seule.
- En appliquant ces règles de transformation d'un diagramme de classe vers un modèle relationnel.

#### ➤ Nous avons abouti un schéma relationnel suivant :



## **Section 2 : CONCEPTION DU SYSTEME**

La phase de conception est une phase essentielle, car elle permet la concrétisation de notre application en s'appuyant sur des modèles de conception définis sur UML. La conception nécessite une analyse rigoureuse et une proposition de modèle adapté pour la réalisation du système. La conception du système comprend deux étapes qui sont la conception générale et la conception détaillée. Dans la conception générale, il faudra définir l'architecture de l'application, le diagramme de packages et le diagramme de déploiement. Pour la conception détaillée, nous allons présenter les différentes classes intervenant et le diagramme de classe.

### **I. Conception générale du système**

Dans cette partie, nous présenterons l'architecture de notre système, le diagramme de packages et le diagramme de déploiement.

#### **1. Architecture du système**

Les architectures basées sur le principe « client-serveur » sont les plus utilisées dans le monde du développement. Ainsi, nous avons tout un ensemble de modèles d'architecture et le plus utilisé est le modèle MVC. Ce modèle reflète même la présentation de l'architecture 3-Tiers.

##### **a. Modèle MVC**

Dans le contexte du développement d'applications mobiles, le Model-View-Controller (MVC) est un modèle architectural et un principe de conception largement adoptés qui favorisent une organisation, une modularisation et une séparation efficaces des préoccupations dans une application. Il offre une approche robuste et structurée du développement d'applications, chaque composant effectuant des tâches spécifiques pour garantir les performances, la maintenabilité et la réutilisabilité du code. L'architecture



MVC est très appréciée en raison de sa polyvalence et de son applicabilité à divers langages de programmation, frameworks et plates-formes.<sup>8</sup>

Le modèle Modèle-Vue-Contrôleur est divisé en trois composants principaux :

2. **Modèle** : le modèle représente la structure des données de l'application et est responsable de la gestion du stockage, de la récupération et de la manipulation des données. Il contient la logique métier de l'application et définit la manière dont les données sont organisées, stockées et gérées, sans interface directe avec l'expérience utilisateur. En séparant la gestion des données de l'interface utilisateur, le modèle garantit que les modifications apportées à un composant n'affectent pas négativement l'autre. Dans le développement d'applications mobiles, le modèle interagit souvent avec des bases de données ou des serveurs distants pour stocker et récupérer les données requises.
3. **View** : le composant View est responsable de l'affichage des données et de l'interface utilisateur (UI) de l'application. Il agit comme un pont entre le modèle et l'utilisateur, en rendant les données dans un format visuellement attrayant et compréhensible. Le composant View ne traite que les éléments de l'interface utilisateur tels que les champs de texte, les boutons, les listes et les images, et ne traite ni ne modifie les données. Dans le développement d'applications mobiles, les vues sont créées à l'aide de composants ou de frameworks d'interface utilisateur natifs tels que Jetpack Compose pour Android et SwiftUI pour iOS.

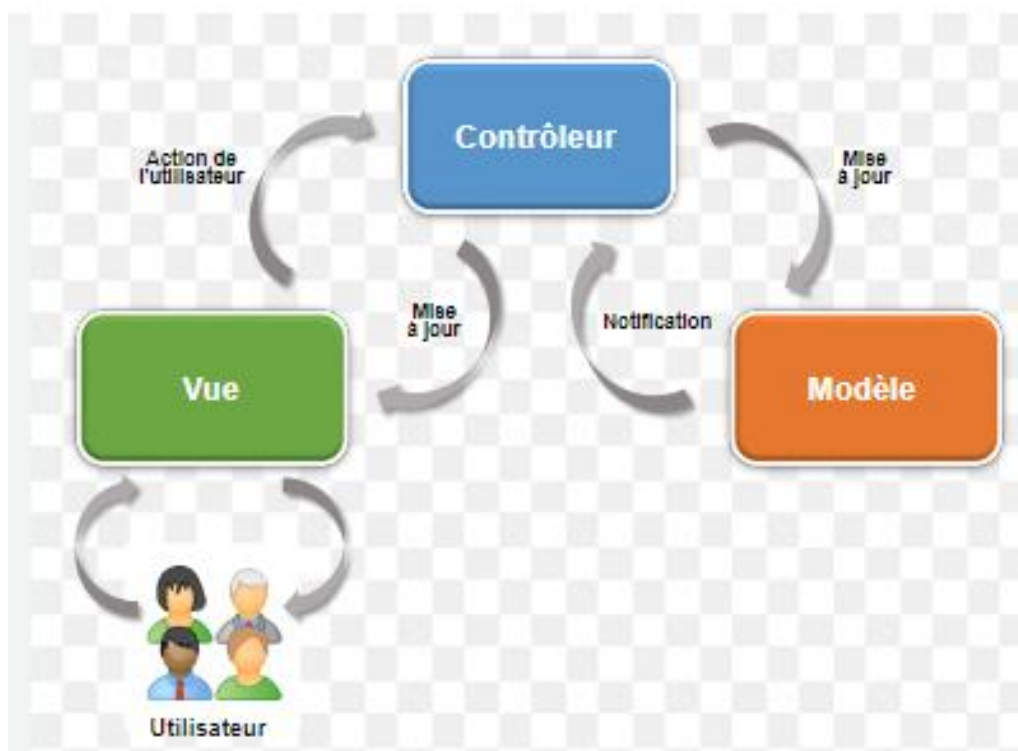
---

<sup>8</sup> <https://appmaster.io/fr/glossary/modele-vue-controleur-mvc>

**4. Contrôleur** : le composant Contrôleur est le ciment qui relie les composants Modèle et Vue. Il agit comme intermédiaire, gérant le flux de données entre eux et gérant les entrées ou les interactions des utilisateurs. Dans le développement d'applications mobiles, le contrôleur est chargé de recevoir les entrées de l'utilisateur via la couche Vue, de traiter ces entrées et d'exécuter la logique métier nécessaire dans la couche Modèle. En fonction du résultat, il met ensuite à jour le modèle et la vue en conséquence, garantissant une expérience utilisateur fluide et réactive.

### 1.1. Présentation du modèle MVC

Beaucoup d'architectures s'appuient sur ce modèle. Le plus connu est donc l'architecture 3Tiers. C'est cette architecture qui sera utilisée pour la réalisation de notre système.

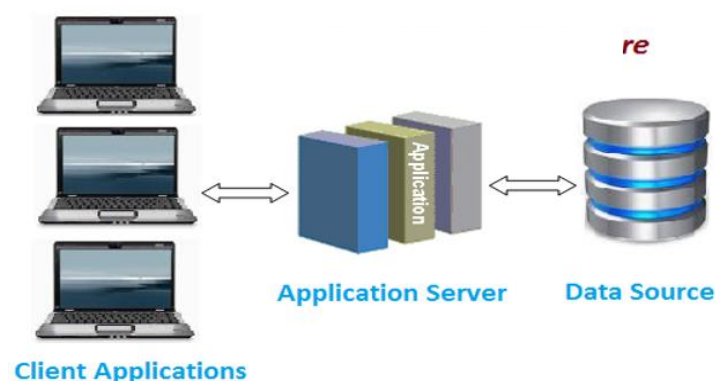


## 1.1. Architecture à 3-Tiers

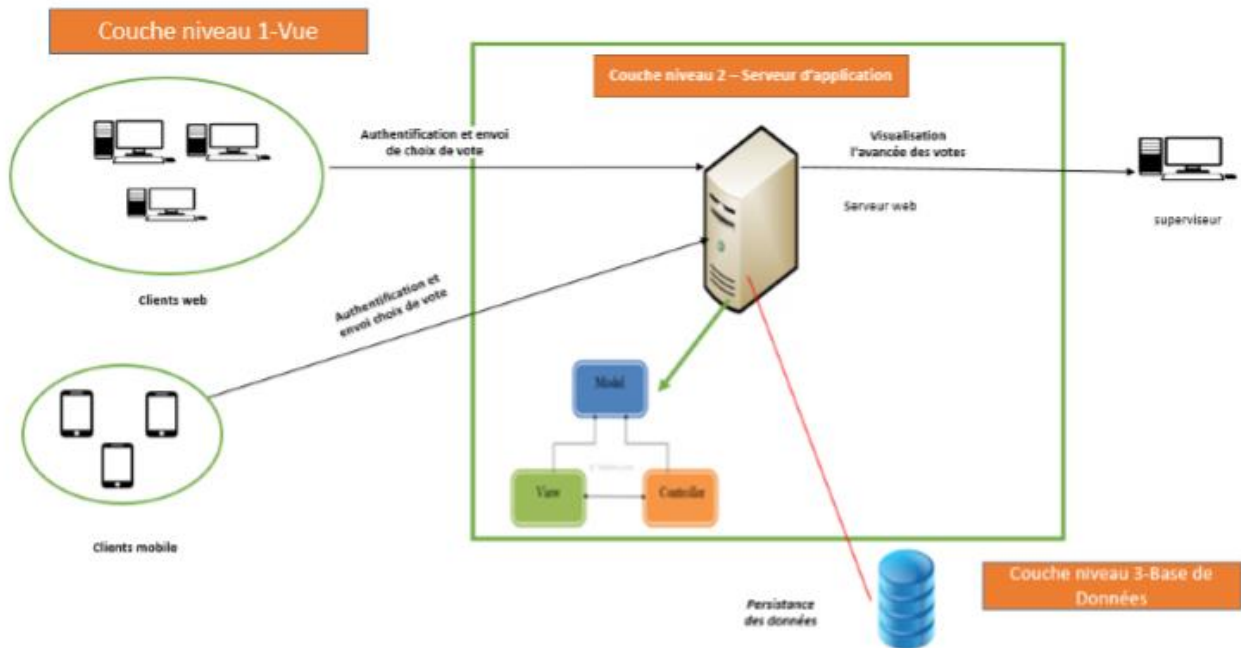
L'architecture trois tiers, également appelée architecture à trois niveaux ou à trois couches, est une architecture client-serveur dans laquelle coexistent et sont maintenus des modules indépendants permettant le rendu d'une interface utilisateur, les procès logiques, fonctionnels et métiers ainsi que l'accès aux données. On parle donc ici d'une infrastructure physique qui va servir de support à une infrastructure logicielle (l'infrastructure trois tiers sous-tend l'infrastructure logicielle). En effet, n'importe quelle application peut être découpée en trois parties : une partie interface graphique, une partie fonctionnelle, et une partie de stockage de données. Et c'est à ces besoins précis que l'architecture trois-tiers s'est dessinée en découpant trois parties distinctes:

### Architecture 3 Tiers

L'architecture 3 tiers nous impose donc à respecter le modèle MVC. L'implémentation de notre système va essentiellement se baser sur ce style architectural.



L'architecture de notre application est représentée par la figure ci-dessous :



## 1.2. Prise en compte de l'accessibilité et de la sécurité des données

### 1.2.1. Accessibilité

La sécurisation de la base des données consiste à restreindre les droits d'accès aux objets et aux relations (de type autre que père et fils) de chaque site organisationnel attribué à chaque personne utilisatrice, c'est-à-dire nous devons limiter les accès à la base à des personnes qui l'utilisent. L'accessibilité aux données se traduit par les actions élémentaires suivantes :

- C : Création d'un enregistrement
- L : Lecture d'un enregistrement
- M : Modification d'un enregistrement
- S : Suppression d'un enregistrement

### **1.2.2. Sécurité**

Les applications mobiles sont devenues un élément clé de la vie quotidienne. Elles nous permettent de communiquer, de travailler et de gérer... Cependant, cette dépendance croissante aux applications mobiles expose également les utilisateurs à des risques de sécurité considérables. Les cybercriminels exploitent les vulnérabilités de ces applications pour accéder aux données personnelles des utilisateurs et peuvent les utiliser à des fins malveillantes.

Dans cet article, nous allons examiner les meilleures pratiques pour la sécurité des applications mobiles.<sup>9</sup>

#### **a. Les risques de sécurité des applications mobiles**

Les applications mobiles sont vulnérables à plusieurs types de menaces de sécurité. Les attaques les plus courantes incluent les attaques par hameçonnage, les attaques de logiciels malveillants et les attaques de réseau. Les vulnérabilités courantes incluent les erreurs de codage, les vulnérabilités du système d'exploitation et les vulnérabilités des protocoles de sécurité. Les conséquences de l'insécurité des applications mobiles peuvent être graves et coûteuses, allant des vols d'identité et des pertes des données aux pertes de confidentialité des données des utilisateurs.

#### **b. Les meilleures pratiques de sécurité des applications mobiles**

Pour protéger les utilisateurs contre les risques de sécurité des applications mobiles, les développeurs doivent adopter les meilleures pratiques de sécurité suivantes :

##### **➤ La sécurité des données :**

les développeurs doivent mettre en place des protocoles de chiffrement pour les données stockées et transmises par l'application. Ils doivent

---

<sup>9</sup> <https://bemobee.com>

également être conscients de la gestion des données personnelles des utilisateurs.

➤ **La sécurité du réseau : Je utiliser le Protocole SSL**

Les protocoles de sécurité réseau tels que **TLS/SSL** pour chiffrer la communication entre l'application et le serveur. L'authentification et l'autorisation doivent aussi être mises en place pour protéger l'application contre les attaques de réseau.

➤ **La sécurité du code :**

Les développeurs doivent mettre en place des pratiques de codage sécurisées et effectuer des tests de sécurité pour détecter et corriger les vulnérabilités de sécurité.

La sécurité du code des applications mobiles est un élément clé pour garantir la protection des données des utilisateurs. Voici quelques mesures à prendre pour assurer la sécurité du code des applications mobiles :

- **Analyse statique du code** : il est important de réaliser une analyse statique du code pour identifier les vulnérabilités potentielles. Cela permet de détecter les erreurs de programmation qui pourraient être exploitées par des attaquants.
- **Analyse dynamique du code** : en plus de l'analyse statique, il est recommandé de réaliser une analyse dynamique du code pour évaluer le comportement de l'application en temps réel. Cela permet de détecter les vulnérabilités liées à la logique de l'application.
- **Vérification de la conformité** : il est important de s'assurer que le code respecte les bonnes pratiques de développement sécurisé, comme l'utilisation de fonctions de chiffrement, la gestion des erreurs et la validation des entrées utilisateurs.

- **Utilisation d'outils de protection du code** : il est possible d'utiliser des outils de protection du code pour empêcher la modification ou la lecture du code. Cela permet de réduire les risques d'attaque liés à la divulgation de secrets de l'application.
- **Formation des développeurs** : les développeurs doivent être formés aux bonnes pratiques de développement sécurisé pour réduire les erreurs de programmation et éviter les vulnérabilités.
- **Tests de sécurité** : il est recommandé de réaliser des tests de sécurité réguliers pour s'assurer que l'application est toujours sécurisée. Cela permet de détecter les vulnérabilités avant qu'elles ne soient exploitées par des attaquants.

➤ **La sécurité de l'utilisateur** :

Les développeurs doivent sensibiliser les utilisateurs aux risques de sécurité des applications mobiles et leur fournir une formation sur les bonnes pratiques de sécurité.

**c. Les outils et technologies de sécurité des applications mobiles**

Il existe plusieurs outils et technologies de sécurité pour les applications mobiles, voici quelques exemples :

- **Analyseur de vulnérabilités** : il s'agit d'un outil qui permet de détecter les vulnérabilités dans les applications mobiles. Il peut scanner les codes sources pour détecter les failles de sécurité et fournir des recommandations pour les corriger.
- **Protection de l'application** : il est possible d'utiliser des outils pour protéger l'application elle-même. Il peut s'agir de solutions de chiffrement pour les données sensibles stockées localement sur l'appareil, ou encore de technologies de protection contre le reverse engineering de l'application.

- **Gestion des identités et des accès** : pour sécuriser les données de l'utilisateur, il est important de mettre en place une solution de gestion des identités et des accès. Cela permet d'authentifier les utilisateurs et de s'assurer que seuls les utilisateurs autorisés ont accès aux données sensibles.
- **Solutions de chiffrement** : le chiffrement est une méthode de protection des données qui consiste à les transformer de telle sorte qu'elles ne puissent pas être lues par des personnes non autorisées. Les solutions de chiffrement sont souvent utilisées pour protéger les données sensibles stockées sur l'appareil.
- **Firewall d'application** : un firewall d'application est une solution qui permet de protéger l'application contre les attaques externes, en bloquant les tentatives d'exploitation de vulnérabilités connues.
- **Solution de détection de malware** : il est important de détecter les malwares qui pourraient être installés sur l'appareil mobile. Des solutions de détection de malware peuvent être utilisées pour identifier les programmes malveillants et les éliminer.
- **Mise à jour régulière** : la mise à jour régulière de l'application est une mesure importante pour garantir sa sécurité. Les mises à jour permettent de corriger les failles de sécurité et d'ajouter des fonctionnalités de sécurité.

#### **D. SYSTEME DE SECURITE CAPTCHA**

Ils sont utilisés principalement pour valider un accès ou une inscription.

Sur Internet, il existe des robots qui scannent en permanence les différentes pages web pour en identifier les ressources. On les appelle des « crawlers » ou « spiders ». La plupart du temps, ils sont utilisés de façon



tout à fait légitime, par les moteurs de recherche ou systèmes d'archivage notamment.

Mais parfois, ils sont détournés pour effectuer des actions malveillantes. Une de ces actions est d'exploiter les formulaires présents sur les sites web (inscription, contact, etc). Le but peut être de trouver des failles ou simplement de créer des comptes en masse, qui pourront être exploités par la suite, notamment dans le cadre de campagne de spams.

C'est pour limiter ces pratiques que les captcha ont été créés. Ils imposent une validation par un humain, ce qui limite l'impact des crawlers.

Ils ont beaucoup évolué en terme de complexité. Dans un premier temps, les captcha se présentaient sous la forme de case à cocher ou de texte simple à recopier, mais les robots, de plus en plus perfectionnés eux aussi, ont été adaptés pour remplir ses validations. Nous avons donc maintenant des images, de textes déformés ou d'objets à identifier, éventuellement plusieurs fois, pour valider les captcha.

Il existe même des captcha invisibles, qui analysent les actions de l'utilisateur sur une page, jusqu'au mouvement de sa souris, pour identifier si l'utilisateur est humain ou non.

### **Un captcha peut-il être trompé ?**

**La réponse est évidente : Oui.**

Nous l'avons dit précédemment, les robots évoluent constamment, les techniques permettent aux programmes de toujours mieux reconnaître les caractères et objets présents sur les images affichées par les captcha. Ils prennent rapidement en compte les nouvelles protections et imitent les comportements attendus par les captcha les plus récents. De plus, ceux-ci présentent, comme tous les programmes informatiques, des failles et des

limites. Ils nécessitent donc un suivi ainsi que des analyses et mises à jour régulières, qui font parfois défaut.

Enfin, des « fermes à captcha » ont été identifiées. Ce sont de véritables usines, situées principalement dans les pays en développement, où le coût de la main d'œuvre est le plus bas. Des équipes de travailleurs humains y remplissent les captcha manuellement, pour des sommes modiques.

Et sans même parler de ces fermes qui industrialisent le processus, un pirate ou un spammeur peut parfaitement valider manuellement un petit nombre de captcha, s'il permet la validation d'un accès sensible.

Le captcha, une mauvaise solution de sécurité ?

Le captcha et ses évolutions ne sont pas de mauvaises solutions, mais ils ne doivent constituer qu'une des briques de la sécurisation de vos systèmes. Se reposer uniquement sur sa validation pour confirmer un accès est une erreur. Il offre un faux sentiment de sécurité qui, nous l'avons vu, peut être facilement contourner. Il n'est justifié qu'en prenant place dans un ensemble de mesures qui œuvrent conjointement à la validation d'une procédure. S'il est la pièce centrale, voir l'unique système de sécurité, il ne peut que constituer une faiblesse dans la chaîne de la cybersécurité.

Chez Altospam, nous utilisons des captcha, notamment au niveau de nos sites Internet, mais nous limitons leur usage et ne les employons que dans des cas très précis et identifiés. En complément de nos très nombreuses mesures de sécurité, cet usage nous semble pertinent. De plus, nous prêtons une attention particulière aux versions et capacités des solutions employées, qu'elles respectent les normes d'accessibilité et ne soient en aucun cas génératrices de blocage de validations légitimes, tout en assurant une sécurité maximum.

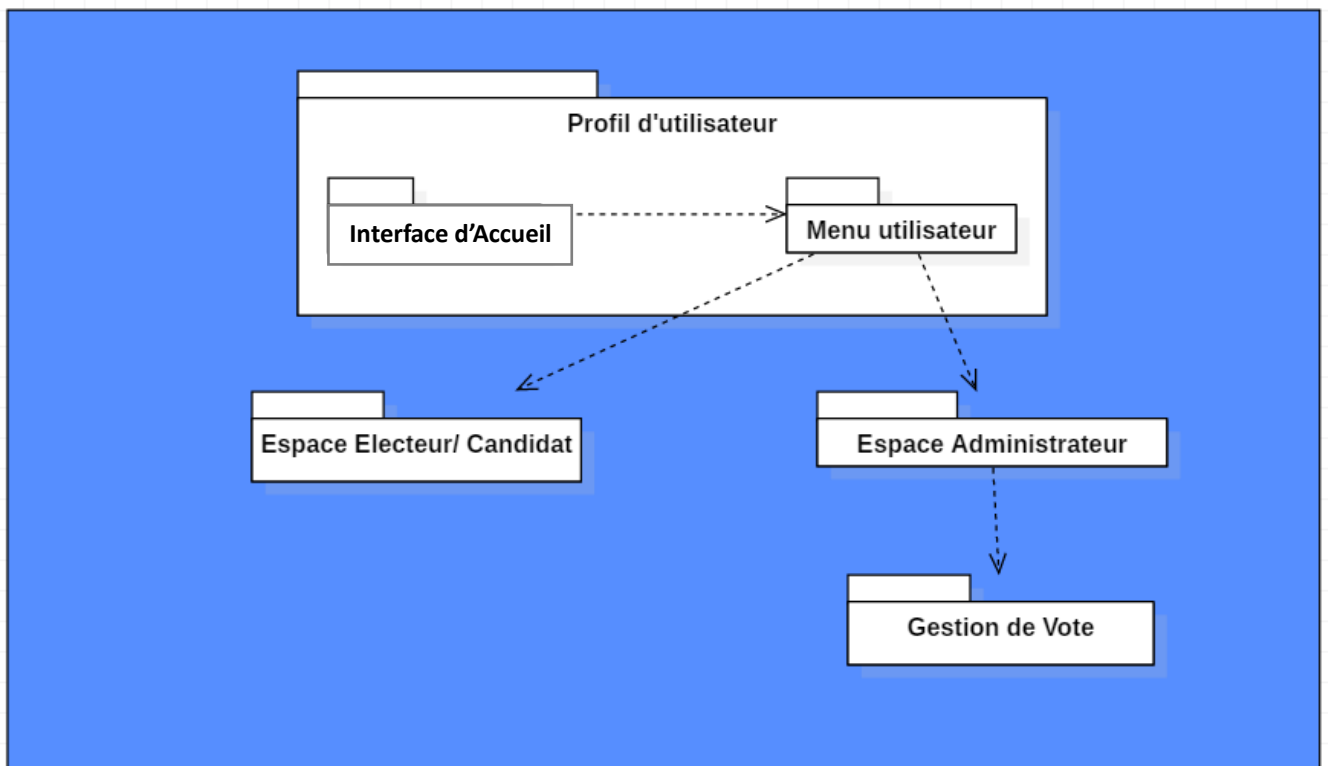
Vous le savez, l'email est le 1er vecteur dans le cadre d'une cyberattaque, il est donc important de sécuriser vos flux d'emails. Chez Altospam, des milliers de clients nous font confiance depuis 20 ans. Tout comme nous l'avons fait avec nos clients, nous sommes toujours prêts à étudier vos problématiques et besoins pour protéger votre structure contre les pirates informatiques et ainsi mieux sécuriser vos données

**En conclusion**, la sécurité des applications mobiles est un sujet crucial dans le monde numérique d'aujourd'hui. Les applications mobiles sont souvent la cible des cybercriminels qui cherchent à exploiter des vulnérabilités dans les applications pour accéder à des informations sensibles ou dérober des données. Pour protéger les utilisateurs contre ces menaces, il est important de mettre en place des meilleures pratiques de sécurité pour les applications mobiles.

## **2. Diagramme de packages**

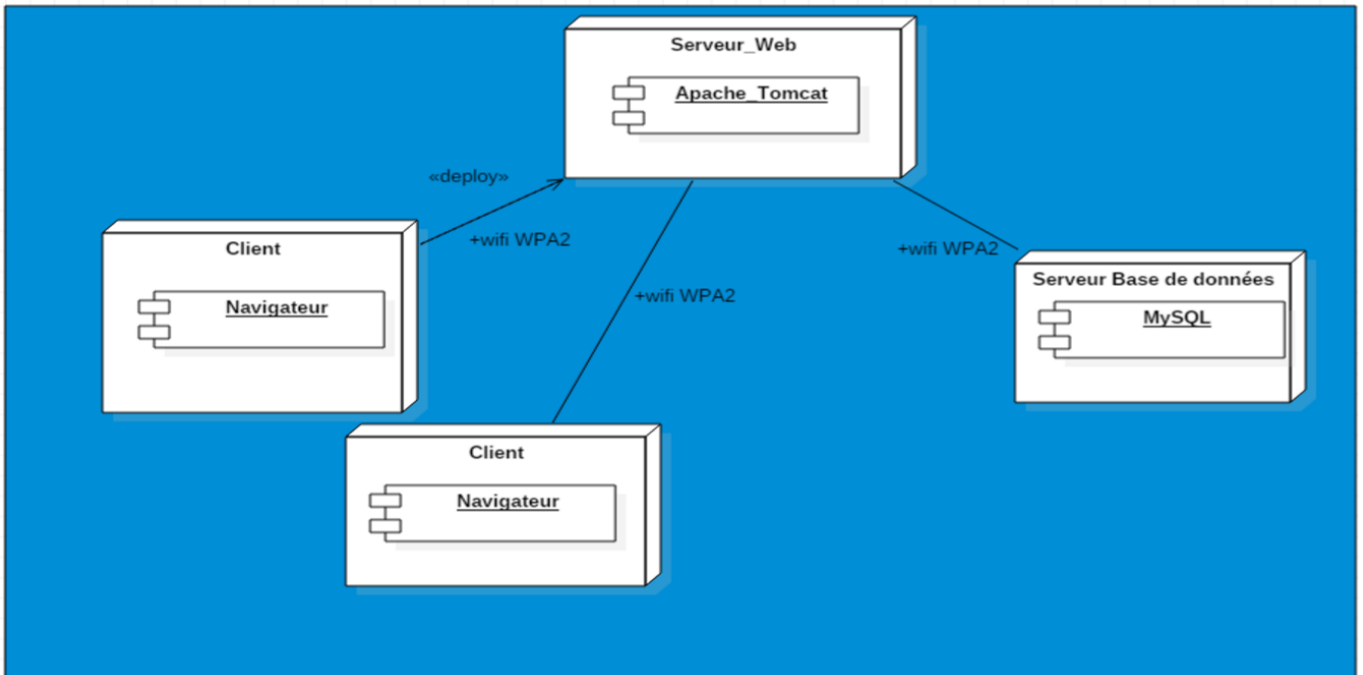
Un diagramme de packages est un diagramme UML qui fournit une représentation graphique de haut niveau de l'organisation de l'application, et vous aide à identifier les liens de généralisation et de dépendance entre les packages. Les packages de notre système sont :

## Diagramme de Packages



### 3. Diagramme de déploiement

En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que les relations entre eux. L'application doit être légère et performante ; elle peut tourner sur des machines de 2Go de RAM et 1.80 GHZ de fréquence processeur. Les ordinateurs tiers pourront accéder au serveur contenu dans une machine d'au moins 8 Go de RAM. La communication est quant à elle assurée par un réseau internet (soit par la technologie Ethernet ou la technologie wifi sécurisé). Le système peut aussi fonctionner en intranet.



## I. Conception détaillée du système

La phase de conception détaillée est la dernière phase nous permettant de faire une représentation statique du système à mettre en place. Le diagramme de classes nous permet de représenter l'ensemble des classes participantes à l'implémentation de notre application. La représentation de notre diagramme s'appuie essentiellement sur les concepts de classes associations. Dans cette partie, nous ferons une large description de nos classes et de notre diagramme de classes.

### 4. Conclusion

Dans ce deuxième chapitre, nous avons effectué et détaillé la conception et la réalisation de notre système de vote présidentiel en s'adaptant en premier lieu sur le langage de modélisation et en se basant sur les diagrammes du formalisme UML, en commençant par le diagramme le plus important qui est celui du cas d'utilisateur et qui sera de plus en plus détaillé en le plus important et qui représente tout notre système et le diagramme de classe.

En deuxième lieu nous avons présenté les règles de passage modèle conceptuel au modèle relationnel.

## **CHAPITRE III IMPLEMENTATION ET PRESENTATION DE L'APPLICATION**

Dans ce chapitre, il sera question d'expliquer les grands points de la réalisation de notre système de vote électronique. L'implémentation va s'appuyer essentiellement sur des résultats fournis dans la partie conception. La réalisation du système implique un choix minutieux des technologies, des langages à utiliser et du type de base de données.

Nous essayerons de détailler la partie implémentation en présentant l'ensemble des outils utilisés et leur application sur notre projet et enfin présenter les principales fonctionnalités de notre application.

### **0. Implémentation de l'application**

Dans cette partie, nous allons présenter les différents outils utilisés pour l'implémentation de notre système.

#### **1.1. Les serveurs utilisés pour notre application**

##### **1.1.1. SGBD (MySQL)**

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de bases de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server. C'est un serveur de bases de données relationnelles développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multi-utilisateur. C'est un logiciel libre open source, développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon c'est la licence publique générale GNU (GPL) qui s'applique. Un logiciel qui

intègre du code MySQL ou intègre MySQL lors de son installation devra donc être libre ou acquérir une licence payante. Cependant, si la base de données est séparée du logiciel propriétaire qui ne fait qu'utiliser des API tiers (par exemple en C# ou PHP), alors il n'y a pas besoin d'acquérir une licence payante MySQL. Ce type de licence double est utilisé par d'autres produits comme le Framework de développement de logiciels.

### 1.1.2. PRESENTATION DE MYSQL

#### ❖ NOTION SUR MYSQL

**MySQL** (prononcer [maj.es.ky.ɛl]) est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde<sup>2</sup>, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

Son nom vient du prénom de la fille du cocréateur Michael Widenius, **My (sv)** (prononcer [my]). SQL fait référence au Structured Query Language, le langage de requête utilisé.

MySQL AB a été acheté le 16 janvier 2008 par Sun Microsystems pour un milliard de dollars américains<sup>3</sup>. En 2009, Sun Microsystems a été acquis par Oracle Corporation, mettant entre les mains d'une même société les deux produits concurrents que sont Oracle Database et MySQL. Ce rachat a été autorisé par la Commission européenne le 21 janvier 2010<sup>4,5</sup>.

Depuis mai 2009, son créateur Michael Widenius a créé MariaDB (Maria est le prénom de sa deuxième fille) pour continuer son développement en tant que projet Open Source.



## ❖ Dénomination

MySQL est l'œuvre d'une société suédoise, MySQL AB, fondée par David Axmark, Allan Larsson et Michael « Monty » Widenius.

Le nom **MySQL** vient de la combinaison de **My**, fille du cofondateur Michael « Monty » Widenius, avec l'acronyme **SQL** (selon la documentation de la dernière version en anglais). D'autre part, le dossier base et une grande partie des bibliothèques utilisées par les développeurs utilisent le préfixe **My**.

Le nom du logo de MySQL (le dauphin) **Sakila**, a été choisi par les créateurs de MySQL sur la base d'un concours. Il a été proposé par Ambrose Twebaze, développeur du Swaziland. D'après Ambrose, le nom Sakila puise ses origines du siswati, la langue locale du Swaziland.

## ❖ DEFINITION DE LA BASE DE DONNEES

Etant un outil communautaire, une base de données est l'ensemble d'information structurées et non redondances.

## ❖ DEFINITION DES TABLES

Une table est l'objet fondamental d'une base de données dans lequel sont enregistré et stockées les données.

Une table possède des lignes dites enregistrements et des colonnes appelées « champ ». Les objets dépendent des tables et chaque table doit avoir une clé primaire.

## ❖ DEFINITION DES FORMULAIRES

Ce sont des interfaces utilisateurs c'est-à-dire interfaces graphiques permettant d'afficher les données en provenance d'une table, ils permettent d'ajouter les données dans une table.

Les formulaires rendent plus conviviale la saisie et la consultation des données.

### **2.1.1. DEFINITION D'ETATS EN SORTIE**

Les états permettent de mettre en forme les résultats des requêtes c'est-à-dire lister sur papier les résultats de l'application. En outre, ils sont des pages d'impression des données ; et son rôle est de présenter des données et de sortir les résultats sur un support imprimable.

#### **a. Serveur d'application (Apache)**

Tomcat est un serveur HTTP à part entière. De plus, il gère les servlets et les JSP. Le JSP est un moteur de template permettant de gérer la vue des données rendues par le Servlet. Tomcat a été écrit en langage Java. Il peut donc s'exécuter via la machine virtuelle Java sur n'importe quel système d'exploitation la supportant.

## **1.1. LANGAGE DE PROGRAMMATION UTILISE DANS NOTRE APPLICATION**

### **A. PHP**

**PHP: Hypertext Preprocessor**, plus connu sous son sigle **PHP** (sigle auto-référentiel), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur web<sup>6</sup>, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook et Wikipédia<sup>8</sup>. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.

## ➤ **Présentation**

PHP est un langage de script utilisé le plus souvent côté serveur : dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendus par un navigateur web. PHP peut également générer d'autres formats comme le WML, le SVG et le PDF.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web. PHP est le plus souvent couplé à un serveur Apache bien qu'il puisse être installé sur la plupart des serveurs HTTP tels que IIS ou nginx. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

C'est un langage peu typé et souple et donc facile à apprendre par un débutant mais, de ce fait, des bugs et des failles de sécurité peuvent rapidement apparaître dans les applications. Pragmatique, PHP ne s'encombre pas de théorie et a tendance à choisir le chemin le plus direct. Néanmoins, le nom des fonctions (ainsi que le passage des arguments) ne respecte pas une logique uniforme, ce qui peut être préjudiciable à l'apprentissage.

## **B. HTML**

HTML est un langage de balisage qui définit la structure du contenu. Un document HTML est une suite d'éléments utilisés pour encadrer différentes parties du contenu afin de les faire apparaître ou se comporter d'une

certaine façon. Les balises autour du contenu permettent de transformer une image ou un texte pour obtenir un lien vers une autre page, de créer des titres, etc.

Les éléments doivent être ouverts et fermés correctement pour contenir ou être contenu. S'ils se chevauchent l'un sur l'autre, votre navigateur web essaiera au mieux de deviner l'intention, mais cela pourra entraîner des résultats inattendus : mieux vaut éviter !

### C. CSS

Le CSS (Cascading Style Sheets ou feuilles de style en cascade) représente un langage informatique dédié à la mise en forme visuelle des documents web.<sup>10</sup>

Conçu pour séparer le contenu d'un site de sa présentation, il confère aux développeurs un **contrôle granulaire** sur l'apparence des éléments HTML. Parmi ses nombreuses fonctionnalités, le CSS permet, entre autres, de gérer l'aspect typographique, la disposition des blocs, les couleurs, les animations et bien d'autres propriétés visuelles incontournables.

#### À quoi sert-il ?

L'utilité première du CSS s'articule autour de trois axes majeurs : l'esthétique, la maintenabilité et l'accessibilité.

Tout d'abord, ce langage ouvre la voie à une infinité de possibilités créatives, permettant d'insuffler vie et personnalité à n'importe quel site web grâce à des styles visuels personnalisés.

---

<sup>10</sup> <https://talks.freelancerepublik.com>

Par ailleurs, en **dissociant la mise en forme du contenu**, le CSS facilite grandement la maintenance des projets en isolant les changements de design dans des fichiers dédiés.

Enfin, les **développeurs front-end** peuvent désormais se conformer aisément aux standards d'accessibilité web en adaptant l'affichage aux différents types d'appareils, navigateurs et même aux besoins des utilisateurs handicapés.

### **Les avantages du CSS pour la conception web**

Au-delà de ses fonctions fondamentales, le CSS offre de nombreux avantages qui en font un atout indéniable pour la conception web :

- **Contrôle visuel avancé** : grâce à sa vaste gamme de propriétés, le CSS permet un contrôle visuel granulaire sur pratiquement tous les aspects d'un site web – des polices aux animations en passant par les dispositions et les effets graphiques complexes.
- **Séparation des préoccupations** : en dissociant le contenu (HTML) de la présentation (CSS), ce langage favorise une meilleure organisation du code et facilite la collaboration entre développeurs et designers.
- **Gain de productivité** : le CSS réduit considérablement les efforts de développement en centralisant la gestion des styles dans des fichiers dédiés, évitant par là même les redondances.
- **Maintenabilité accrue** : comme les modifications de design s'effectuent de manière centralisée, la maintenance est, de fait, garantie sur le long terme des projets web.

- **Compatibilité multiplateforme** : bien sûr, le CSS permet d'adapter l'affichage aux différents navigateurs, résolutions d'écran et types d'appareils.

## 1.2. API

Interface de programmation d'application (API)

### a) Qu'est-ce qu'une API ?

API est l'acronyme de Application Programming Interface (interface de programmation d'application), un code de programmation qui relie les programmes informatiques entre eux. En termes simples, il s'agit de tout type d'interface logicielle qui permet à deux programmes différents de communiquer.<sup>11</sup>

Les API ont augmenté la connectivité entre une large gamme de programmes et ont donné naissance à diverses solutions et applications en ligne dans tous les secteurs et dans toutes les langues. De plus, les API ouvrent la voie à de nouvelles activités et favorisent le progrès grâce à l'intégration.

En fait, beaucoup d'entre nous utilisent des API au quotidien sans même s'en rendre compte. Par exemple, lorsque vous consultez les prévisions météorologiques sur votre téléphone, une API relie l'application de votre téléphone au réseau météo. Lorsque vous vous connectez à Facebook ou Instagram via Google, vous utilisez une API. Des institutions financières comme les banques et les fournisseurs de cartes de crédit aux services de streaming en ligne comme Netflix et Spotify, les solutions API tierces améliorent notre expérience en ligne collective.

---

<sup>11</sup> <https://www.wix.com/encyclopedia>

## b) Types d'API

**Voici les quatre types d'interfaces de programmation d'applications Hybride :**

- **Les API ouvertes** désignent tout type d' interface open source à laquelle les développeurs et les utilisateurs peuvent avoir accès. Ce type d'API peut également être appelé API publique, qui, comme son nom l'indique, est partagée librement et publiée ouvertement en ligne pour que tout le monde puisse l'utiliser.
- **Les API partenaires** ne sont pas ouvertes au public et nécessitent des autorisations spécifiques pour y accéder ou les intégrer. Ces types d'API sont plus stratégiques et servent à connecter les entreprises B2B et B2C.
- **Les API internes**, également appelées API privées, ne sont pas ouvertes aux utilisateurs externes. Ces types d'API restent exclusifs à une entreprise spécifique et ne sont utilisés que pour améliorer les communications et le développement internes, comme les systèmes de paie ou de RH.
- **Les API composites** sont une combinaison d'au moins deux API ou plus. Ce type d'API est unique car il peut accéder à plusieurs systèmes en un seul appel. Les API composites sont capables de rassembler des requêtes provenant de plusieurs sources dans une séquence organisée, en un seul appel d'API.

### 1.3. Technologie

Une multitude de solutions pour développer une application mobile existent. Il est donc important de comprendre laquelle est la plus adaptée selon son besoin. Ces solutions peuvent être classées en trois catégories : Natif, Web cross-platform et Natif cross-platform.<sup>12</sup>

#### Quelques définitions

- Une **application native** est une application développée pour se conformer entièrement aux exigences et aux directives d'un système d'exploitation particulier.
- Une **application web** est accessible via un navigateur web et hébergée par un serveur.
- Une **application hybride** combine l'application native et web.
- Le terme de **cross-platform** est utilisé pour une application qui peut être utilisée sur plusieurs plateformes.

### 1.4. Choix de la Technologie <sup>13</sup>

Notre choix de la technologie est les applications hybride

Une application hybride est une application logicielle qui combine des éléments d'applications natives et d'applications Web.<sup>14</sup>

Les applications hybrides sont essentiellement des applications Web qui ont été placées dans un shell d'application natif.

---

<sup>12</sup> <https://makina-corpus.com>

<sup>13</sup> Note de cours L2 Programmation mobile , op.cit

<sup>14</sup> <https://apresta.fr>



Une fois qu'il est téléchargé à partir d'un magasin d'applications et installés localement, le shell est capable de se connecter à toutes les fonctionnalités de la plate-forme mobile via un navigateur intégré à l'application.

Le navigateur et ses plug-ins s'exécutent sur le back-end et sont invisibles pour l'utilisateur final.

### **Les développeurs privilégient les applications hybrides**

Les applications hybrides sont populaires car elles permettent aux développeurs d'écrire du code pour une application mobile une fois tout en prenant en charge plusieurs plates-formes.

Étant donné que les applications hybrides ajoutent une couche supplémentaire entre le code source et la plate-forme cible, elles peuvent fonctionner légèrement plus lentement que les versions natives ou Web de la même application.

Les développeurs de logiciels se tournent de plus en plus vers des outils de test automatisés et des flux de travail pour accélérer les schémas de test tout en garantissant une meilleure cohérence et exhaustivité du processus d'assurance qualité, mais le succès exige un haut niveau de compréhension.

### **Caractéristiques des applications hybrides**

Les fonctionnalités des applications hybrides comprennent :

- La capacité de fonctionner que l'appareil soit connecté ou non.
- Intégration avec le système de fichiers de l'appareil mobile.
- Intégration avec les services Web.

- Un navigateur intégré pour améliorer l'accès au contenu dynamique en ligne.

## **Fonctionnement les applications hybrides**

Les applications hybrides fonctionnent de la même manière que les applications Web, mais comme les applications natives, elles sont téléchargées sur l'appareil.

À l'instar des applications Web, les applications hybrides sont généralement écrites en HTML5, CSS et JavaScript. Les applications hybrides exécutent du code dans un conteneur.

Le moteur de navigateur de l'appareil est utilisé pour afficher les API HTML et JavaScript et natives pour accéder au matériel spécifique à l'appareil.

Bien qu'une application hybride partage généralement des éléments de navigation similaires en tant qu'application Web, le fait que l'application puisse ou non fonctionner hors connexion dépend de ses fonctionnalités.

Si une application n'a pas besoin de la prise en charge d'une base de données, elle peut alors fonctionner hors ligne.

**Tout projet de conception digitale nécessite un développement front end et back end.** Le premier s'occupe de ce qui est visible pour les internautes. Quant au second, il se charge d'organiser les processus d'arrière-plan.

Ce projet nécessitera l'intervention d'un développeur **front end** chargé de concevoir l'interface utilisateur destinée aux Electeur et candidat, et

d'un développeur **back end** qui s'occupera de réaliser l'interface administrateur destinée à l'administrateur.

## **A. Le front end : travailler sur ce que l'internaute voit**

Le développeur front end a pour mission de programmer l'ensemble des éléments visibles par les utilisateurs/visiteurs d'un site web ou d'une application mobile. Il s'occupe donc de l'interface graphique et de l'ergonomie.<sup>15</sup>

Son objectif est d'offrir aux utilisateurs une expérience satisfaisante. Dans cette optique, il crée et met en place les éléments graphiques qui leur permettront de se retrouver sur l'interface et de la prendre facilement en main. Il s'assure aussi que celle-ci reste ergonomique.

Ce programmeur intervient aussi bien sur la mise en page que sur les boutons, les menus déroulants, les formulaires ou les liens internes. Son travail comprend également la gestion du responsive design, des animations et des interactions.

En outre, le développeur front end collabore étroitement avec au moins un webdesigner et idéalement un expert UX.

### **Les langages de programmation utilisés en front end**

Les premiers outils d'un développeur front end sont les langages de programmation qu'il utilise pour écrire ses lignes de code. Leur maîtrise est indispensable. Les langages front end principaux sont :

- HTML : ce langage permet de créer et d'organiser le contenu web de manière claire (polices, couleurs, etc.) ;

---

<sup>15</sup> <https://blog.lesjeudis.com>

- CSS : il complète les fonctions du HTML et organise la présentation de la plateforme ;
- Javascript : il sert à insérer des éléments interactifs tels que les menus déroulants et les formulaires contacts pour dynamiser les pages.

## **Les frameworks et bibliothèques front end**

Pour améliorer leur efficacité, les développeurs front end se reposent sur les frameworks. Ce sont des squelettes de code pré-écrit dans lesquels ils introduisent du codage spécifique au domaine pour aller plus vite. Parmi les frameworks front end les plus utilisés, nous avons :

- Angular JS ;
- Backbone ;
- **Bootstrap** ;
- React.JS.

Enfin, les programmeurs se servent aussi des bibliothèques ; il s'agit de paquets de code pré-écrit qu'ils peuvent insérer dans leur propre code. Lodash, **jQuery** et LESS sont les bibliothèques front end les plus utilisées.

Il existe un grand nombre de bibliothèques et de frameworks que vous pouvez consulter sur le guide l'Open Source. Cela vous permettra de découvrir leurs spécificités et leurs atouts.

En tout, le développement front end se concentre sur la partie visible, c'est-à-dire l'interface que l'internaute voit et utilise au quotidien. Mais une belle interface ou web design reste une coquille vide si une équipe de développement back end n'intervient pas sur le site ou l'application.

## **B. LE BACK END : TRAVAILLER SUR CE QUE L'INTERNAUTE NE VOIT PAS**

Contrairement au développeur front end, le développeur back end se concentre sur la partie immergée de l'iceberg. Il opère en coulisses. Sa responsabilité est de développer les fonctionnalités du site web ou de l'application mobile en cours d'élaboration.<sup>16</sup>

Le travail du développeur back end est essentiellement invisible pour l'utilisateur final. Cela n'enlève néanmoins rien à son importance. Son intervention s'articule autour de trois points.

Le premier est le serveur d'hébergement. Ici, le programmeur back end doit élaborer un espace de stockage qui permettra d'enregistrer tous les fichiers qui composent le projet, y compris les éléments qui définissent sa partie visible.

La création d'une base de données est le deuxième point. Elle permet de conserver dans des tableaux spécifiques toutes les données nécessaires au bon fonctionnement du site ou de l'application (ID d'un utilisateur, droits d'utilisateurs, stockage de mots de passe).

Enfin, le développeur web back end s'occupe de la mise en place de l'application web. Il devra dans cette optique concevoir toutes les fonctionnalités qui permettront au site de fournir les informations demandées à l'utilisateur. Son travail inclut également les éléments suivants

:

- La communication entre les serveurs et les applications par le biais d'interfaces de programmation d'applications (API) ;

---

<sup>16</sup> <https://blog.lesjeudis.com>

- La logique back-end ;
- La conformité aux politiques d'accessibilité et de sécurité.

## **Les langages de programmation utilisés en back end**

Comme on peut s'y attendre, le développement back end requiert une palette d'outils de programmation différents de celle utilisée en front end. De ce fait, le programmeur back end va utiliser les langages suivants :

- **PHP** : c'est un langage de script côté serveur utilisé dans le développement web ;
- Java plus connu sous le nom de *loosely coupled programming language* (langage de programmation à couplage lâche) : il peut être exécuté sur toutes les plateformes compatibles avec ce type de langage ;
- Python : c'est un langage puissant, complet et très démocratisé. Il est idéal pour l'écriture de scripts et le développement d'applications sur la plupart des plateformes ;
- C++ : il est très utilisé pour la programmation concurrentielle.

## **Les frameworks et bibliothèques back end**

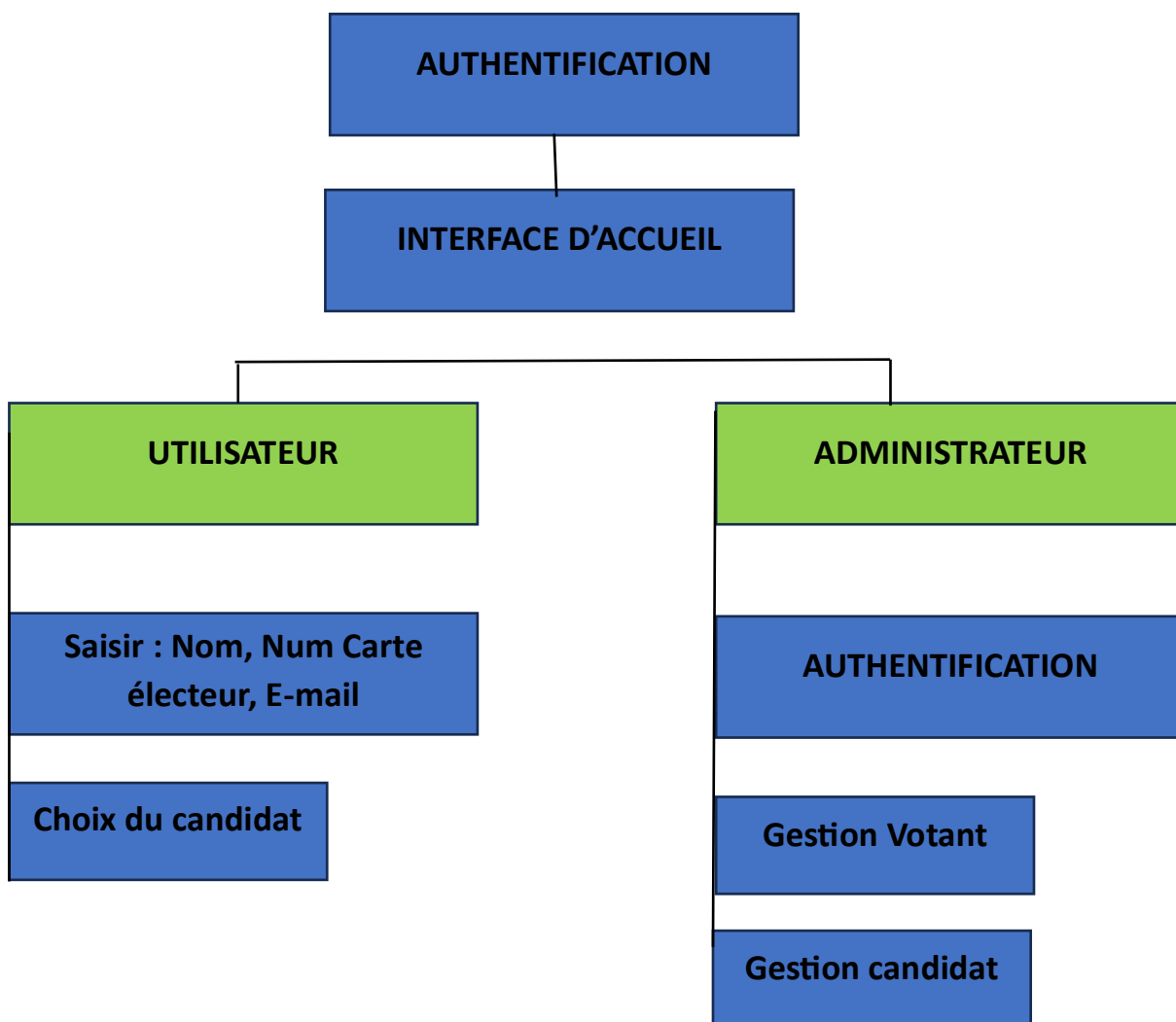
Pour les mêmes raisons de coût, rapidité et rentabilité que le développeur front end, le développeur back end utilise aussi des frameworks. Sans eux, les fonctions courantes comme le défilement infini ne seraient pas possibles. Parmi les plus populaires, on peut citer :

- Symfony ;
- Django ;
- Ruby on Rails ;
- Angular ;
- Meteor.

Grâce à l'essor d'Angular, le développement PHP n'est pas attribué qu'au développement back end. Ainsi, un programmeur back end peut s'orienter vers du développement en JS.

Enfin, les programmeurs back end utilisent aussi des bibliothèques, afin de bénéficier d'une meilleure flexibilité. Les bibliothèques Python comme FastApi, Pilow et TensorFlow sont les plus appréciées.

### 1.5. ARCHITECTURE PHYSIQUE DU SYSTEME



## 2. PRESENTATION DE NOTRE APPLICATION

Un projet simple basé sur un système de vote qui utilise le langage PHP. Le projet PHP suivant contient toutes les fonctionnalités essentielles qui peuvent être utilisées pour les étudiants en informatique de première et deuxième année pour leurs projets universitaires. Il possède des fonctionnalités importantes qui permettront à tous les utilisateurs d'interagir de la même manière que les systèmes de vote interagissent avec les électeurs sur la base du choix du bon candidat. Ce système ainsi que le concept de l'application web sont clairs, c'est le même que les scénarios de la vie réelle et bien mis en œuvre sur celui-ci.

### A. PAGE D'ACCUEIL



The screenshot shows the homepage of a mobile voting system. At the top, there is a dark navigation bar with the text "SYSTEME DE VOTE MOBILE" and "Features" on the left, and a green "Admin Panel" button on the right. The main content area is light gray and features a large black icon of a hand pointing to a speech bubble containing the word "VOTE". Below the icon, the text "SYSTEME DE VOTE MOBILE" is displayed in a large, bold, sans-serif font. Underneath this, the tagline "Sûr . Fiable . Sécurisé . Rapide" is written in a smaller font. At the bottom, there is a red button with a white ballot icon and the text "Donnez vos votes maintenant".



## B. INTERFACE DE VOTE

### ELECTION PRESIDENTIELLE

#### Choisissez un candidat!

---

nom complet de l'électeur :

Email enregistré de l'électeur :

Numéro de carte d'électeur :

Selectionnner un candidat:

- Félix ANTOINNE TSHISEKEDI TSHILOMBO
- Moise KATUMBI CHAPWE
- Martin FAYULU MADIDI
- Samy BADIBANGA
- Vital KAMERHE

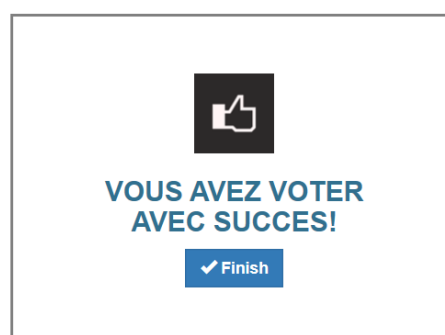
---

Confirmer votre vote
Decline

Activer Windows  
Accédez aux paramètres p

HP/index.html

## C. VOTE REUSSIT



## D. INFO SUR LES CANDIDATS

Lorsque cette page est ouverte vous allez écouter hymne Nation de la RDC.

### SYSTEME DE VOTE MOBILE



**Félix ANTOINNE TSHISEKEDI TSHILOMBO**

Félix ANTOINNE TSHISEKEDI TSHILOMBO, né le 13 juin 1963 à Léopoldville (aujourd'hui Kinshasa, est un homme d'Etat congolais, président du Congo depuis le 25 janvier 2019

- Age: (61 ans)
- Pays: RDC
- Niveau d'étude: Gradué en Marketing et communication
- Parti politique: UDPS



**Moise KATUMBI CHAPWE**

Moise KATUMBI CHAPWE, né le 28 décembre 1964 à Kashobwe dans le haut-katanga, est un homme d'affaires, homme politique et dirigeant sportif congolais

- Age: (59 ans)
- Pays: RDC
- Niveau d'étude: Master en Criminologie
- Parti politique : ENSEMBLE POUR LA REPUBLIQUE



**Martin FAYULU MADIDI**

Martin FAYULU MADIDI, né le 21 novembre 1956 à Kinshasa, est un homme politique de la République Démocratique du Congo (RDC).

- Age: (67 ans)
- Pays: RDC
- Niveau d'étude: Master en administration des affaires
- Parti politique : ECIDE



**Samy BADIBANGA**

Samy BADIBANGA, né le 12 septembre 1962 à Kinshasa, est un homme d'Etat kino-congolais.

- Age: (62 ans)
- Pays: RDC
- Niveau d'étude: Diplome d'Etude Supérieurs à Anvers
- Parti politique : LES PROGRESSISTES



**Vital KAMERHE**

Vital KAMERHE, né à Bukavu le 04 mai 1959 dans la province Kivu, est un polyglotte qui maîtrise globalement les 4 langues nationales du Congo : le Kikongo, le Lingala, le Swahili et le Tshiluba en plus de certaines langues étrangères l'anglais.

- Age: (65 ans)
- Pays: RDC
- Niveau d'étude: Doctorat en économie
- Parti politique : UNC

## E. PAGE AUTHENTIFICATION ADMINISTRATEUR

### Login Admin

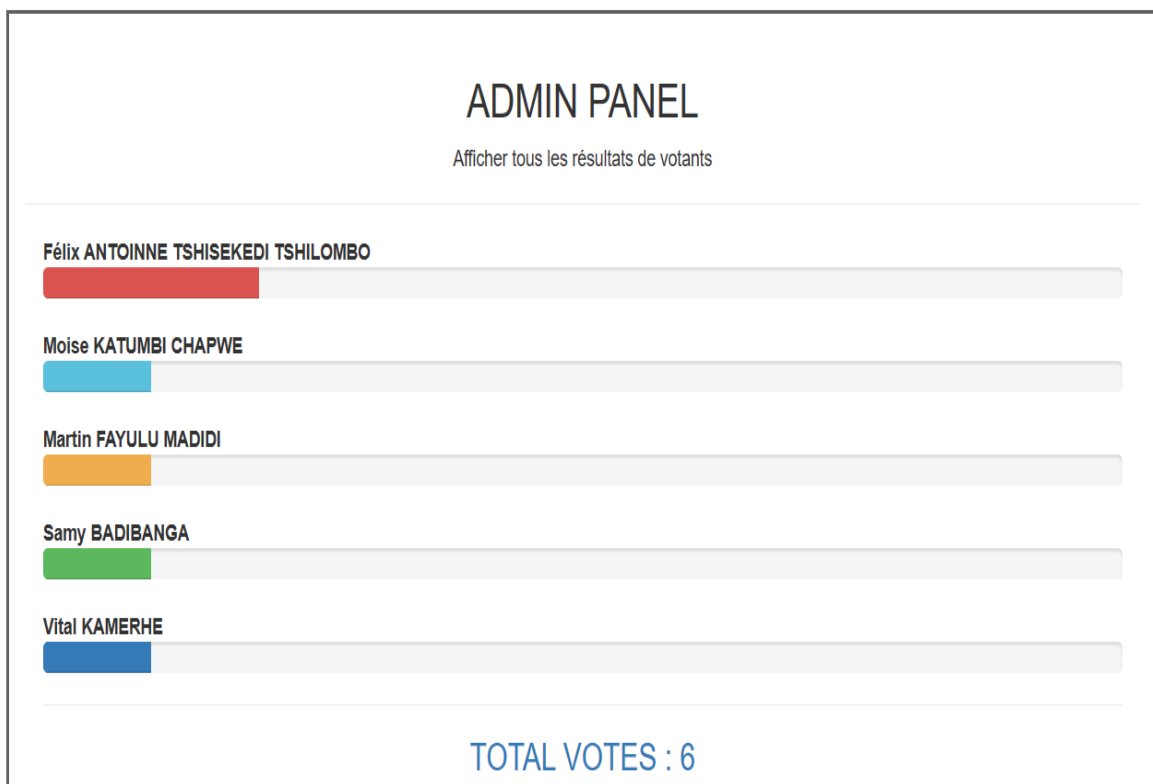
---

**Username**

**Password**

[LOGIN](#)

## F. PANEL ADMIN (AFFICHER TOUS LES RESULTATS DE VOTANS)



## G. LISTE DES ELECTEURS DEJA VOTER

### SYSTEME DE VOTE MOBILE

Afficher les électeurs qui ont déjà voter

nom complet de l'électeur	E-Mail	Numéro de carte d'électeur
KONGA MASSAMBA Théophile	massambatheophile01@GMAIL.com	31984463
NGONDA MASSAMBA Kerlin	ngondakerlin@gmail.com	2147483647
		0
KALONDA MANGA Christenvie	kalondachristenvie@gmail.com	2147483647
MULUMBA MASANKA Lajoie	mulumbalajoie@gmail.com	2147483647
MAKANGILA TONNY Sarah	makangilasarah@gmail.com	2147483647
TSHIMANGA BOOTA Camile	camile0100@gmail.com	2147483647

## H. CHANGER MOT DE PASSE

### SYSTEME DE VOTE MOBILE

### Admin's Password

---

**Old Password**

**New Password**

**Retype Password**

## CONCLUSION GENERALE

Nous voici au terme de notre étude qui apporté sur « **SYSTEME DE VOTE MOBILE** » **CAS DE LA COMMISSION ELECTORA**, afin de permettre à celle-ci une bonne organisation des élections à courte durée par une application qui sera mise à sa disposition.

**En effet, nous avons parcourus 3 Chapitre pour atteindre notre But :**

- Le premier chapitre était basé sur l'étude des besoins fonctionnels

Dans ce chapitre on a parlé de l'ensemble des besoins qu'ils soient fonctionnels et non fonctionnels.

Le vote est le moyen d'expression central d'une démocratie. Il existe dans toutes les sphères du pays au sein des associations et des entreprises, dans les universités et, bien évidemment, au niveau politique. Le développement des technologies de l'information semble pouvoir modifier la physionomie et le déroulement du vote.

C'est ici que nous avons étudié notre champ d'investigation ; en commençant par la présentation de la Commission Electorale Nationale Indépendante, ainsi que de sa structure en se limitant à un domaine de précis, nous l'avons critiqué afin de voir ses défaillances et faire les suggestions et en fin c'était de notre obligation de proposer une solution. L'informatique étant un outil important qui rend le monde plus petit avec l'évolution de la technologie et présente plusieurs avantages avec la solution de la technologie hybride (mobile), nous avons proposé à la CENI la solution mobile pour une élection efficace qui facilitera le développement de la CENI.

- Le second chapitre était basé sur la conception du nouveau système d'information

Dans ce dernier, notre but était la conception d'une base de données par la méthode d'analyse MERISE qui est une méthode de conduite des projets d'informatiques.

➤ La dernière partie était basée sur la réalisation de l'application

Sachant que la limite n'était pas seulement la conception d'une base de données, il a été question de recourir à la programmation pour mettre en place une application. Pour y arriver, nous avons recouru au langage de programmation Javascript et PHP avec le langage de balisage HTML, pour programmer notre application et le SGBD Mysql pour implémenter notre base de données.

Nous sommes disposés à toutes les remarques et suggestions, aux lecteurs qui liront ce travail car étant humain et la science étant vaste.

Après avoir tout réalisé, nous disons que notre projet est atteint.

## **BIBLIOGRAPHIE**

### **I. Notes Cours**

- Note de cours G2 Méthode de Recherche Scientifique, op.cit
- Ibidem
- MALANDA, Cours de MAI2,G3 info0,UWB, 2021-2022

### **II. WEBIOGRAPHIE**

- <https://www.Commençamarche.com>
- Ibidem
- Ibidem
- <https://www.Wikipedia.com>
- <https://prezi.com/p/2ikdh0zd8x4f/historique-de-uml/>
- <https://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours01.html>
- <https://appmaster.io/fr/glossary/modele-vue-controleur-mvc>
- <https://bemobee.com>
- <https://talks.freelancerepublik.com>
- <https://www.wix.com/encyclopedia>
- <https://makina-corpus.com>
- <https://apresta.fr>
- <https://blog.lesjeudis.com>
- <https://blog.lesjeudis.com>

## Table des matières

<b>EPIGRAPHIE</b> .....	i
<b>DEDICACE</b> .....	ii
<b>REMERCIEMENTS</b> .....	iii
<b>0. INTRODUCTION GENERALE</b> .....	1
<b>1. CHOIX ET INTERET DU SUJET</b> .....	2
<b>1.1. CHOIX DU SUJET</b> .....	2
<b>1.2. INTERET DU SUJET</b> .....	2
<b>2. DE LIMITATION DU SUJET</b> .....	3
<b>2.1. DANS LE TEMPS</b> .....	3
<b>2.2. DANS L'ESPACE</b> .....	3
<b>3. SUBDIVISION DU TRAVAIL</b> .....	3
<b>CHAPITRE I SPECIFICATION ET ANALYSE DES BESOINS</b> .....	4
<b>I. Introduction :</b> .....	4
<b>I.1. Etude de l'existant :</b> .....	4
<b>I.1.1. Historique</b> .....	4
➤ <b>Présidents</b> .....	4
<b>I.1.2. SITUATION GEOGRAPHIQUE</b> .....	5
<b>I.1.3. Mission</b> .....	5
<b>I.1.4. Moyen humain (Services)</b> .....	5
<b>I.1.5. Moyen matériel</b> .....	6
<b>I.1.6. Organigramme de la CENI</b> .....	7
<b>1.2. Description de l'existant :</b> .....	8
<b>1.2.1. Historique du vote électronique</b> .....	8
<b>1.2.2. Critique de l'existant :</b> .....	10
<b>1.3. Problèmes rencontrés sur l'organisation des élections</b> .....	12
<b>1.3.1. Le temps et le travail fastidieux sur l'organisation des élections</b> .....	12
<b>1.3.2. Problème de transparence et de fiabilité</b> .....	13
<b>1.4. Problématique du sujet</b> .....	14
<b>1.5. Solution proposée :</b> .....	14
<b>1.6. Objectifs spécifiques</b> .....	15
<b>1.7. Etude des besoins</b> .....	16
<b>1.7.1. Besoins fonctionnels :</b> .....	17
<b>1.7.2. SPECIFICATION DES BESOINS</b> .....	17
<b>1.7.3. Tableau d'ordonnement des tâches</b> .....	21
<b>1.7.4. Besoins non fonctionnels :</b> .....	22



1.8. Cadre du projet :	22
<b>CHAPITRE II ETAPE CONCEPTUELLES</b>	24
II.1. Introduction :	24
<b>Section 1 : Présentation de langage UML :</b>	24
A. Introduction :	24
2.2. PRESENTATION DES DIAGRAMME	25
2.2.1. Diagramme de cas d'utilisation	25
<b>3. Description des cas d'utilisation</b>	28
3.1. Analyse des besoins fonctionnels du système	33
3.2. Etude des besoins fonctionnels du système	33
3.3. Représentation du diagramme de séquence	33
3.4. Description des classes	39
#	39
3.4.1. Présentation des règles de gestion	39
3.4.1.1. Règles d'action (RA)	39
3.4.2. Recensement et description des relations	40
a. Recensement des relations	40
3.5. Diagramme de Classe	40
Concepts et instance	41
Encapsulation	41
3.5.1. Présentation de Diagramme de Classe	42
<b>Section 2 : CONCEPTION DU SYSTEME</b>	44
<b>I. Conception générale du système</b>	44
<b>1. Architecture du système</b>	44
1.1. Présentation du modèle MVC	46
1.2. Prise en compte de l'accessibilité et de la sécurité des données	48
1.2.1. Accessibilité	48
1.2.2. Sécurité	49
a. Les risques de sécurité des applications mobiles	49
b. Les meilleures pratiques de sécurité des applications mobiles	49
c. Les outils et technologies de sécurité des applications mobiles	51
<b>D. SYSTEME DE SECURITE CAPTCHA</b>	52
2. Diagramme de packages	55
3. Diagramme de déploiement	56
<b>I. Conception détaillée du système</b>	57
<b>CHAPITRE III IMPLEMENTATION ET PRESENTATION DE L'APPLICATION</b>	59

<b>O. Implémentation de l'application</b> .....	59
<b>1.1. Les serveurs utilisés pour notre application</b> .....	59
<b>1.1.1. SGBD (MySQL)</b> .....	59
<b>1.1.2. PRESENTATION DE MYSQL</b> .....	60
❖ <b>Dénomination</b> .....	61
❖ <b>DEFINITION DE LA BASE DE DONNEES</b> .....	61
❖ <b>DEFINITION DES FORMULAIRES</b> .....	61
<b>2.1.1. DEFINITION D'ETATS EN SORTIE</b> .....	62
<b>1.1. LANGAGE DE PROGRAMMATION UTILISE DANS NOTRE APPLICATION</b> .....	62
➤ <b>Présentation</b> .....	63
<b>À quoi sert-il ?</b> .....	64
<b>Les avantages du CSS pour la conception web</b> .....	65
<b>1.2. API</b> .....	66
<b>1.3. Technologie</b> .....	68
<b>Quelques définitions</b> .....	68
<b>1.4. Choix de la Technologie</b> .....	68
Notre choix de la technologie est les applications hybride.....	68
<b>Les développeurs privilégient les applications hybrides</b> .....	69
<b>Caractéristiques des applications hybrides</b> .....	69
<b>Fonctionnement les applications hybrides</b> .....	70
<b>A. Le front end : travailler sur ce que l'internaute voit</b> .....	71
<b>Les langages de programmation utilisés en front end</b> .....	71
<b>Les frameworks et bibliothèques front end</b> .....	72
<b>B. LE BACK END : TRAVAILLER SUR CE QUE L'INTERNAUTE NE VOIT PAS</b> .....	73
<b>Les langages de programmation utilisés en back end</b> .....	74
<b>Les frameworks et bibliothèques back end</b> .....	74
<b>1.5. ARCHITECTURE PHYSIQUE DU SYSTEME</b> .....	75
<b>2. PRESENTATION DE NOTRE APPLICATION</b> .....	76
<b>CONCLUSION GENERALE</b> .....	81
<b>BIBLIOGRAPHIE</b> .....	83