

UNIVERSITE LIBRE DES PAYS DES GRANDS LACS

FACULTE DE SCIENCES ET TECHNOLOGIES

DEPARTEMENT DE GENIE ELECTRIQUE ET INFORMATIQUE



BP. 368 GOMA

www.ulpgl.net

**CONCEPTION ET REALISATION D'UNE
APPLICATION WEB POUR LA GESTION DE
VOTE DES REPRESENTANTS DES
ETUDIANTS AU SEIN DE L'ULPGL GOMA**

Par : **RUBENI BENJAMIN**

Travail présenté et défendu en vue de l'obtention du
Diplôme de Licencié en Sciences et Technologies
Appliquées

Option : **Génie Informatique**

Directeur : **Dr. Ir BARAKA MUSHAGE Olivier**

Encadreur : **M. Ir NZANZU VINGI Patrick**

ANNEE ACADEMIQUE 2024 - 2025

Epigraphe

« L'accessibilité numérique est un impératif démocratique, chaque voix compte et chaque vote doit être rendu possible »

Haben Girma

Dédicace

A mes très chers parents **RUBENI GAFURURA Dominic** et **MUJAWIMANA MAOMBI Sophia**.

RUBENI BENJAMIN

Remerciements

Le résultat d'innombrables efforts et sacrifices des plusieurs personnes ont aidé à la réalisation de ce présent travail qui marque la fin de nos études universitaires à la faculté des sciences et technologies à l'université libre des pays des grand-lacs (ULPGL) de Goma.

Nos remerciements s'adressent en premier lieu au maitre de temps et des circonstances lui qui permet que nous soyons toujours en vie malgré la situation sécuritaire que nous traversons et sans lui nous ne pourrions jamais être capable de réaliser quoi que ce soit.

En second lieu nous tenons à dire un mot de remerciement aux autorités de l'université libre des pays des grands lacs de nous avoir donné une si bonne formation, en particulier nous remercions notre Directeur le **Pr. Dr. Ir Olivier BARAKA MUSHAGE** et notre Encadreur le Master **Ir NZANZU VINGI Patrick** pour leurs conseils et orientations tout au long de la rédaction de notre travail.

Nos remerciements s'adressent également à nos très chers parents **RUBENI GAFURURA Dominic** et **MUJAWIMANA MAOMBI Sophia** qui ont mis toute leur énergie pour que nous puissions bien approfondir nos connaissances intellectuelles et ainsi nous garantir un bon avenir.

Nos remerciements s'adressent aussi à nos frères et sœurs : **MUZUNGU, USHINDI, CLAUDE, DANIEL, FABRICE, SAIDI, SANDRINE, JOYEUSE**, pour les soutiens tant moraux qu'intellectuel et pour leur encouragement.

Nos remerciements s'adressent aussi à nos camarades de L3 FST de l'année académique 2024-2025 pour leur courage et leur détermination depuis l'année préparatoire jusqu'aujourd'hui, en particulier je remercie mes camarades : Alfred, Danny, Théophile, Jonathan, Yves et bien d'autres que je ne saurais pas citer dans ce paragraphe.

Nous remercions également nos amis et connaissances qui d'une manière ou d'une autre ont contribué à la réalisation de ce travail, qu'il ressente en ces mots la gratitude de nos remerciements les plus sincères.

RUBENI BENJAMIN

Résumé

Le choix des représentants des étudiants influence sur le déroulement académique, la passation des cours, des examens et cela a un grand impact sur la réussite des étudiants.

C'est en guise de cela que nous avons trouvé nécessaire de mettre en place un système de vote fiable et rapide qui permettrait à tous les étudiants de participer au vote et d'exprimer librement leur choix.

Le cœur de ce travail réside dans la réalisation d'une application web fonctionnelle qui permettra de résoudre le problème de vote au sein de l'ULPGL Goma.

Pour réaliser ce travail nous avons recouru aux idées et avis des autres étudiants qui nous ont parlé des défauts et inconvénients que présente le système archaïque avec bulletin de vote qui serait vulnérable aux fraudes mais aussi qui est trop lent.

Grâce à quelques technologies apprises à l'auditoire entre autres, le html, le css, JavaScript, php, mysql, python et tant d'autres, nous avons pu mettre en place un système qui facilitera le vote et qui est rapide, fiable, robuste et sécurisé.

*Mots clés : **html, vote, css, JavaScript, php, mysql***

Abstract

The choice of student representatives influences on the academic routine, the delivery of courses and exams, and even has a significant impact on student success.

Reason why we found that it's necessary to implement a fast voting system that would allow all students to participate to the vote and freely express their choice.

This work is focused on the development of a functional web application that would solve the voting problem within ULPGL Goma.

In order to realise this work, we gather ideas and opinions of other students who told us about the defects and disadvantages of the archaic voting system, which is vulnerable to fraud and also too slow.

Trought some technologies learned in the auditorium, including HTML, CSS, JavaScript, PHP, MySQL, and many others, we were able to implement a fast, reliable, robust, and secure system that would facilitate voting.

Keywords: HTML, vote, CSS, JavaScript, PHP, MySQL "

Table des matières

Epigraphe.....	i
Dédicace	ii
Remerciements	iii
Résumé	iv
Abstract.....	iv
Table des matières	vi
Liste des sigles et abréviations	ix
Liste des tableaux	xi
Liste des figures.....	xii
0. Introduction générale.....	1
0.1 Contexte/Généralités sur le thème.....	1
0.2 Identification et formulation du problème.....	1
0.3 Questions de recherche.....	2
0.4 Formulation des hypothèses	2
0.5 Justification du choix du sujet et motivations	3
0.6 Énoncé des objectifs de recherche	3
0.6.1. L'objectif général	3
0.6.2. Les objectifs opérationnels/spécifiques.....	3
0.7 Méthodologie et technique utilisée	4
0.7.1 Méthodologies	4
0.7.2 Techniques	5
0.8 Délimitation du travail	5

0.9 Subdivision du travail.....	6
Chapitre 1	7
REVUE DE LA LITTERATURE ET GENERALITES SUR LE VOTE ELECTRONIQUE.....	7
1.1 Revue de la littérature.....	7
1.1.1 Problèmes des systèmes de vote traditionnels.....	7
1.1.2 État de l'art et solutions existantes.....	10
1.1.3 Aspects techniques et méthodologiques de la conception	11
1.1.4 Tableau synthétique des systèmes de vote déjà conçu	12
1.2 Généralités sur le vote électronique.....	16
1.2.1. Historique du vote électronique	16
1.2.2. Types de votes électroniques.....	17
1.2.3 Avantages et inconvénients des systèmes de votes électroniques.....	18
Conclusion partielle.....	19
Chapitre 2	21
APPROCHE CONCEPTUELLE DU SYSTEME DE VOTE.....	21
2.1 Analyse du système de vote.....	21
2.1.1 Analyses des besoins et contraintes du système.....	21
2.2 Présentation des technologies à utiliser.....	22
2.2.3. Le langage de modélisation.....	24
2.3 Cahier des charges détaillé du projet.....	24
2.3.1 Tableau de spécifications fonctionnelles.....	26
2.3.2 Tableau des spécifications non fonctionnelles	29
2.4 Présentation des différents diagrammes de modélisation.....	30
2.4.1 Diagramme de cas d'utilisation.....	31
2.4.2 Diagrammes de séquence du processus de vote.....	40
2.4.3 Diagramme de classe.....	45
2.4.4 Diagramme de déploiement	47

Conclusion partielle.....	48
IMPLEMENTATION	50
3.1 Environnement de développement et cadre de réalisation	50
3.2 Architecture technique et organisation de l'application.....	51
3.3 Implémentation du module d'inscription	52
3.4 Implémentation du module de gestion des candidatures.....	53
3.5 Implémentation du module de gestion des scrutins.....	54
3.6 Implémentation du module de vote électronique	56
3.7 Calcul automatique et affichage des résultats	57
3.8 Sécurité du système et protection des données	58
3.9 Tests, validation et fiabilité du système	58
Conclusion partielle.....	59
Conclusion générale	60
Difficultés rencontrées	61
Améliorations possibles et perspectives.....	62
Recommandations	62
Bibliographie	63
Annexes	64

Liste des sigles et abréviations

BP : boîte postale

CSS: cascading style sheets

CV: curriculum vitae

Dr: Docteur

E-mail: electronic mail

EVM: electronic voting machine

e-voting : electronic voting

FSTA : Faculté des Sciences et des Technologies Appliquées

HTML: Hyper Text Mark Up Language

INRIA : Institut National de Recherche en Informatique et en Automatique

Ir: ingénieur

i-voting: internet voting

Js: javascript

L3: licence3

Mysql: My structured query language

PHP: hypertext preprocessor

PIN: personal identification number

Pr : professeur

RSA : rivest shamir adleman

SHA-256: secure hash algorithm 256 bits

SMS: short message service/sent

SQL: structured query language

UASZ : Université Assane Seck de Ziguinchor

ULPGL : Université Libre des Pays des Grands Lacs

UML: unified modeling language

UX: user experience

Vs code: visual studio code

XAMPP: cross-platform apache Mysql PHP Perl

XOR : exclusive OR

XSS : Cross-site Scripting

Liste des tableaux

Tableau 1. Tableau synthétique des systèmes de vote déjà conçu	12
Tableau 2 Tableau de spécifications fonctionnelles.....	26
Tableau 3 Tableau des spécifications non fonctionnelles	29
Tableau 4 documentation du CU inscription et connexion	35
Tableau 5 documentation du CU importation de la liste de étudiants.....	35
Tableau 6 documentation du CU Soumission d'une candidature	36
Tableau 7 documentation du CU suivi du statut de la candidature.....	36
Tableau 8 documentation du CU validation ou rejet d'une candidature.....	37
Tableau 9 documentation du CU création d'un vote.....	37
Tableau 10 documentation du CU attribution des candidats a une élection	37
Tableau 11 documentation du CU consultation des élections disponibles	38
Tableau 12 documentation du CU voter.....	38
Tableau 13 documentation du CU confirmer vote	39
Tableau 14 documentation du CU calcul et affichage des résultats.....	39
Tableau 15 documentation du CU générer et diffuser résultats	40

Liste des figures

Figure 1 html	22
Figure 2 css.....	22
Figure 3 javascript	23
Figure 4 php	23
Figure 5 sql.....	23
Figure 6 UML.....	24
Figure 7 architecture client-serveur pour application web de vote	26
Figure 8 diagramme de cas d'utilisation	34
Figure 9 diagramme de la séquence inscription	41
Figure 10 diagramme de la séquence soumission de candidature.....	42
Figure 11 diagramme de la séquence création élection.....	43
Figure 12 diagramme de la séquence vote	44
Figure 13 diagramme de la séquence proclamation des résultats	45
Figure 14 diagramme de classe	47
Figure 15 diagramme de déploiement	47
Figure 16 page d'inscription et de connexion	52
Figure 17 page validation de candidature	53
Figure 18 page soumission de candidature	54
Figure 19 page création d'une élection	55
Figure 20 page assignation des candidats a une élection	56
Figure 21 page de vote	57
Figure 22 page affichage des résultats du vote	58

0. Introduction générale

0.1 Contexte/Généralités sur le thème

Au regard de l'ampleur croissante des avancées technologiques et de leur évolution continue, il devient essentiel d'en exploiter le potentiel afin d'améliorer la rentabilité, l'efficacité et la rapidité dans l'exécution des activités courantes.

Notons que la conception et la réalisation d'un système de vote a déjà fait l'objet d'attention dans le passé, de la part de la communauté scientifique entre autres dans le travail de :

1. **YOUSFI IZZEDDINE**, travail portant sur « **la conception et la réalisation d'un système de vote à distance** » étudiant de l'université Mohamed Boudiaf en Algérie.

2. **ADAM SEYE**, travail portant sur « **la mise en œuvre d'un système de vote électronique, applications aux élections organisées à l'UASZ** ».

Aujourd'hui l'université libre des pays des Grands Lacs, avec ses plus de 3000 étudiants, fait face à des défis majeurs dans ses élections de représentants des étudiants suite à un processus de vote manuel fastidieux, avec le temps d'exécution trop long et un manque de transparence menaçant la crédibilité des résultats du vote. Pour y remédier, nous avons développé une application web de vote en ligne. Ce système sécurisé permettra aux étudiants de voter de n'importe où et n'importe quand, garantira la sécurité des votes grâce au chiffrement, et assurera un décompte instantané et fiable. L'objectif est d'accroître massivement la participation, d'optimiser les ressources et de restaurer la pleine démocratie étudiante de l'université.

C'est en guise de cela en tant qu'aspirant ingénieur que nous avons songé à chercher une solution technologique pour améliorer le système de gestion de vote au sein de l'ULPGL Goma.

0.2 Identification et formulation du problème

Dans le contexte académique, l'élection des représentants des étudiants représente un élément fondamental de la démocratie au sein de l'université. Malgré leur importance, ces processus électoraux sont fréquemment conduits de manière conventionnelle, en utilisant des bulletins en

papiers, des registres manuels et des comptages prolongés. Cela rend le processus vulnérable aux erreurs humaines, aux contestations, à la fraude et même à des retards dans l'annonce des résultats. Dans le contexte de la digitalisation, les technologies mobiles et web présentent des possibilités inédites pour moderniser ces élections. Le développement d'une application web spécialisée pourrait à la fois faciliter le processus électoral et assurer la transparence, l'efficacité et la sûreté des votes.

0.3 Questions de recherche

Dans cette perspective, nous nous sommes proposés les questions de recherche suivantes

1. Comment est-ce qu'un système informatique de vote améliore-t-il la gestion de vote ?
2. Comment élaborer et déployer une application web sécurisée, performante et fiable pour l'organisation et la gestion efficace des élections estudiantines à divers échelons, tout en assurant la transparence et la participation active ?
3. Quels sont les facteurs qui influencent l'adoption et l'utilisation d'un système informatique de vote par les étudiants et le personnel universitaire ?

0.4 Formulation des hypothèses

Lors de cette recherche, nous formulons plusieurs hypothèses pour orienter notre exploration dans la création d'un système informatique de vote et pour répondre aux questions de recherche énoncées précédemment.

L'implémentation d'une application web pour le vote améliorerait la gestion de vote en permettant un dépouillement automatique, en évitant les attroupements des étudiants dans la salle de vote, mais aussi un système informatique permettrait la gestion de vote d'une manière rapide et fiable et cela augmenterait le taux de participation des étudiants aux élections de leurs représentants comparativement aux élections précédentes faites avec un système de vote traditionnel.

L'élaboration et le déploiement de cette application reposeraient sur l'adoption d'une architecture client-serveur associée à des technologies web performantes et des protocoles de sécurité avancés, permettant ainsi de garantir la fiabilité du système et une haute disponibilité pour les électeurs. Par ailleurs, les facteurs qui favorisent l'adoption et l'utilisation d'un système informatique de vote

sont : la rapidité du processus, l'automatisation des tâches fastidieuses et l'augmentation de taux de participation des électeurs.

0.5 Justification du choix du sujet et motivations

Il y a de cela quelques années que nous sommes étudiants au sein de l'université libre des pays des Grands Lacs dans la faculté des sciences et des technologies et pendant tout ce temps, l'organisation du vote des représentants des étudiants se faisait d'une manière archaïque et cela nous a poussés à vouloir apporter une contribution en développant une application web qui faciliterait le processus de vote vu que le procédé archaïque de vote présente plusieurs inconvénients.

C'est en guise de cela que nous avons choisi ce sujet en raison de sa pertinence potentielle sur l'amélioration des votes des représentants des étudiants, en exploitant les opportunités offertes par les nouvelles technologies.

Le choix de ce sujet pour nous a été motivé par le fait que nous voulions apporter une valeur ajoutée à l'amélioration de la vie estudiantine en concevant une application web qui permettrait la passation de vote d'une manière aisée, rapide et bien sécurisée.

0.6 Énoncé des objectifs de recherche

0.6.1. L'objectif général

L'objectif principal de ce travail est de concevoir un système informatique pour la gestion des votes de représentants des étudiants au sein de l'ULPGL.

0.6.2. Les objectifs opérationnels/spécifiques

Les objectifs spécifiques de notre travail sont les suivants :

- Analyser les besoins fonctionnels et non fonctionnels du système de vote électronique en milieu académique, en tenant compte des contraintes de sécurité, de fiabilité et d'accessibilité.
- Concevoir une architecture web adaptée permettant une séparation claire entre les couches (présentation, logique métier, base de données) tout en garantissant la scalabilité et la robustesse du système.

- Développer une interface utilisateur intuitive et responsive, facilitant l'accès au système depuis différents terminaux (ordinateurs, tablettes, smartphones).
- Mettre en œuvre des mécanismes d'authentification et de sécurisation des données, notamment le chiffrement des identifiants, la protection contre les fraudes électorales et les attaques informatiques.
- Intégrer un système de gestion des rôles et permissions pour différencier les droits des étudiants votants, des administrateurs, et des agents de supervision.
- Implémenter un algorithme de dépouillement automatique, garantissant la transparence, la traçabilité et l'exactitude des résultats électoraux en temps réel.
- Documenter l'ensemble du processus de développement, de la phase de spécification jusqu'à la mise en production, afin d'assurer la maintenabilité et l'évolutivité du système.

Ces objectifs spécifiques orienteront la recherche et la méthodologie de notre étude pour répondre de manière approfondie aux questions de recherche posées ci-haut.

0.7 Méthodologie et technique utilisée

Pour mener à bien notre travail de conception d'un système de vote étudiant, nous avons adopté quelques méthodologies et techniques qu'on utilisera tout au long de notre travail.

0.7.1 Méthodologies

Méthodologie de l'approche centrée sur l'utilisateur : avec cette méthode nous comptons récolter des informations, des avis des utilisateurs entre autres les étudiants, les candidats et l'administration pour savoir leurs besoins et les transformer en spécifications fonctionnelles et non fonctionnelles pour notre application avant même d'entamer le code.

Méthodologie agile : cette méthode nous permettra de faire un développement itératif et incrémental, avec des essais fréquentes et de versions fonctionnelles et une grande flexibilité pour s'adapter aux retours des utilisateurs.

0.7.2 Techniques

Techniques d'entretien : avec cette technique nous comptons organiser quelques entretiens avec l'administration pour avoir assez d'informations sur les règlements de vote, les contraintes légales, les exigences de sécurité, les systèmes d'identification des étudiants, la liste des étudiants et la procédure des proclamations des résultats.

Au-delà de cela nous comptons également organiser des entretiens avec les étudiants pour savoir leurs habitudes d'utilisation d'application, leurs besoins en matière de transparence, de communication avec les électeurs.

Nous comptons également organiser un entretien avec les représentants des étudiants sortant pour comprendre les défis du vote physique.

Technique documentaire : cette technique nous permettra de consulter quelques documents par rapport à notre travail, entre autre les travaux de nos aînés qui concerne le même sujet que nous ou soit un sujet similaire au notre, à part cela nous comptons consulter les règlements intérieurs de l'université en ce qui concerne les élections étudiantes, nous comptons aussi consulter quelques documentation sur les applications existants qui gère également le vote au sein d'une université et en déterminer les failles, enfin nous consulterons les manuels de procédures de vote existants pour y identifier toutes les étapes pouvant être informatiser.

0.8 Délimitation du travail

Nous ne pouvons dire que nous irons jusqu'à étudier tous les aspects et processus de vote mais plutôt nous pouvons dire que la solution que nous proposons ne fait que résoudre en partie le problème lié au vote des représentants des étudiants vu que nous sommes délimités d'une manière temporelle et d'une manière intellectuelle également.

Notre solution est conçue pour L'ULPGL mais dans l'avenir nous pourrions envisager de l'adapter de sorte qu'il soit utilisé pour tous les instituts supérieurs et universités.

Dans le cadre de ce travail nous nous limiterons à un système de vote fait que pour l'ULPGL.

0.9 Subdivision du travail

Mis à part l'introduction générale et la conclusion, ce travail sera subdivisé en trois chapitres qui sont :

Chapitre 1. Revue de la littérature et Généralités sur le système de vote électronique

- Exploration des autres travaux similaires au nôtre
- Introduction approfondie par rapport aux systèmes de vote électronique
- Types de systèmes de votes électronique

Chapitre 2. Approche conceptuelle du système de vote

- Présentation des technologies à utiliser
- Présentation des différents diagrammes de modélisation

Chapitre 3. Implémentation

- Présentation des fonctionnalités et du design du système
- Test

Ces trois chapitres formeront le cœur de notre travail et donneront l'aspect général de notre solution.

Chapitre 1

REVUE DE LA LITTERATURE ET GENERALITES SUR LE VOTE ELECTRONIQUE

Le présent chapitre a pour objectif d'établir les fondations théoriques et contextuelles de notre travail en effectuant une revue de la littérature approfondie et en posant les généralités sur le vote électronique.

Dans un premier temps, nous procéderons à une revue de la littérature qui mettra en lumière les problèmes inhérents aux systèmes de vote traditionnels tout en dressant un état de l'art des solutions et protocoles existants.

Cette analyse permettra également de dégager les aspects techniques et méthodologiques cruciaux pour la conception d'une application de vote fiable.

Dans un second temps, nous présenterons les généralités sur le vote électronique. Cette section retracera l'historique de cette technologie, des premières machines des années 1960 à l'adoption du vote électronique.

Enfin, nous synthétiserons les avantages et les inconvénients qui sont essentiels pour orienter le développement de notre propre solution.

Ce chapitre est ainsi fondamental pour comprendre le contexte du problème et justifier les choix technologiques et méthodologiques qui seront développés dans les chapitres suivants.

1.1 Revue de la littérature

1.1.1 Problèmes des systèmes de vote traditionnels

Plusieurs chercheurs se sont intéressés à la modernisation des systèmes électoraux à travers l'utilisation des technologies numériques afin de répondre aux problèmes de lenteur, de fraude et de manque de transparence observés dans les méthodes de vote traditionnelles. Dans cette

perspective, **MUHIRE BUUNDA Victor** [1] a réalisé un travail intitulé « Conception et mise en œuvre d'un système de vote électronique : application aux élections de l'ULPGL ». L'objectif de son étude était de remplacer le vote manuel par une plateforme numérique permettant aux étudiants de participer aux élections universitaires à distance. Le système développé offre plusieurs fonctionnalités telles que le dépôt des candidatures en ligne, leur validation, le vote électronique ainsi que la consultation des résultats. Cette solution a permis d'améliorer la rapidité du processus électoral et de réduire les contraintes liées au vote manuel. Toutefois, le système repose sur une architecture centralisée qui ne garantit pas totalement l'intégrité des données et la protection contre d'éventuelles manipulations des résultats.

Dans le même domaine, **MUGISHO CIRINDYE Frederick** [2] a proposé un système de vote en ligne intégrant la technologie blockchain afin de renforcer la sécurité et la transparence des élections. Son étude met en évidence les limites des systèmes électoraux traditionnels en République Démocratique du Congo, notamment les risques de fraude, les erreurs de dépouillement et les retards dans la publication des résultats. Pour répondre à ces problèmes, l'auteur a utilisé la blockchain comme mécanisme d'enregistrement sécurisé et immuable des votes. Cette approche améliore considérablement la traçabilité et la fiabilité des opérations électorales. Néanmoins, son travail est davantage orienté vers les élections à grande échelle et n'aborde pas suffisamment certains aspects liés à l'authentification avancée des électeurs ou à l'intégration de technologies intelligentes d'identification.

Adama Seye dans son mémoire [3] rapporte que, l'organisation d'élections classiques, comme celles tenues à l'Université Assane Seck de Ziguinchor (UASZ), est une tâche fastidieuse qui demande beaucoup de ressources humaines et matérielles. Il note également un faible taux de participation et la difficulté de garantir la transparence des résultats, en raison du dépouillement manuel qui est source d'erreurs. Ces problèmes soulignent la nécessité de moderniser le processus électoral pour le rendre plus efficace et plus fiable.

En 1995 dans son article [4], Michael J. Radwin propose un protocole de vote électronique visant, à introduire l'intraçabilité et la commodité dans les systèmes existants, tels que ceux de Sako-Kilian et Gennaro [5]. Le travail s'appuie sur la technique des signatures aveugles (blinded signatures) de David Chaum [6] pour garantir qu'il est "impossible pour quiconque de tracer le bulletin de vote jusqu'à l'électeur". En appliquant les signatures aveugles, l'électeur obtient un pseudonyme

numérique signé. Ce mécanisme préserve l'anonymat de Victor l'électeur et permet de le démasquer uniquement s'il tente de voter deux fois (double-vote), en utilisant les propriétés de l'opération XOR (ou exclusif) sur les réponses à un défi aléatoire.

Le protocole atteint les objectifs suivants :

- **Confidentialité**
- **Vérifiabilité universelle** : Cette propriété permet à toute partie intéressée de vérifier la bonne exécution de l'élection
- **Intraçabilité**
- **Commodité** : en permettant potentiellement à l'électeur de voter depuis n'importe quel emplacement en réseau (téléphone, GAB, etc.), sans exiger sa présence à un bureau de vote désigné ou une carte à puce.

Cependant, Radwin conclut que l'atteinte de la vérifiabilité universelle par la publication des pseudonymes et des votes sacrifie la propriété de non-reçutage. Ce compromis signifie que l'électeur garde un reçu de son vote, ce qui rend le protocole vulnérable à la coercition ou à la vente de votes.

Au Canada, en 2010, un rapport [7] portant sur « l'analyse comparative du vote électronique » met un accent sur l'évaluation fondamentale des facteurs à considérer pour l'introduction du vote à distance par Internet dans le contexte électoral canadien. Le document souligne la nécessité de trouver un équilibre délicat entre l'augmentation de l'accessibilité et la garantie d'une sécurité et d'une intégrité maximales.

L'analyse met en lumière plusieurs avantages potentiels du vote par Internet, notamment l'amélioration de l'accessibilité pour les personnes à mobilité réduite ou celles en déplacement, ainsi que l'accroissement de la participation électorale, un phénomène observé lors de divers essais municipaux au Canada.

Cependant, le rapport identifie des risques majeurs qui soulèvent des questions pour la littérature :

- **Sécurité et confidentialité** : Les risques de cyberattaques, d'erreurs de chiffrement, et de compromission du secret du vote sur des appareils non sécurisés sont jugés préoccupants.

- **Intégrité du vote** : La vulnérabilité à la coercition (pression exercée sur l'électeur dans l'environnement privé du domicile) et à la fraude (usurpation d'identité) demeure un obstacle significatif pour l'adoption de cette méthode.
- **Équité** : L'éventuel renforcement du « clivage numérique » risque de désavantager les électeurs ayant un accès limité ou coûteux à Internet.

Bien que l'environnement canadien soit jugé favorable à l'adoption de cette technologie, le rapport insiste sur le fait qu'il n'existe pas de modèle étranger directement transposable. Il recommande fortement une approche progressive et prudente pour le Canada, commençant par une définition précise des besoins et la réalisation d'essais à petite échelle afin de construire un modèle national qui ne sacrifie ni la sécurité ni l'intégrité démocratique au nom de la seule commodité. Ce travail souligne ainsi que le vote par Internet est une question de compromis technique et sociétal plutôt qu'une simple solution technologique.

L'analyse de ces différents travaux montre une évolution progressive des systèmes de vote, passant de la simple numérisation du processus électoral à l'intégration de mécanismes avancés de sécurité et d'identification. Toutefois, malgré les avancées réalisées, certaines limites persistent en matière de sécurité globale, d'authentification fiable, de protection des données et d'adaptation aux besoins spécifiques des utilisateurs. C'est dans cette optique que s'inscrit notre recherche, qui vise à proposer une solution plus complète répondant aux insuffisances observées dans les travaux antérieurs tout en garantissant davantage de sécurité, de transparence, de fiabilité et d'accessibilité.

1.1.2 État de l'art et solutions existantes

La littérature parcourue présente différentes solutions et expériences de vote électronique à travers le monde, mettant en lumière des systèmes déjà en place et des cadres théoriques.

1. Systèmes académiques : Le mémoire de Yousfi Izzeddine [8] mentionne deux systèmes open source reconnus dans la recherche :
2. Helios Voting [9] : Un système de vote en ligne de bout en bout vérifiable, qui est utilisé par des institutions universitaires pour des élections. Sa conception permet à tout électeur de vérifier que son bulletin a bien été pris en compte sans compromettre l'anonymat.
3. Belenios [10] : Un autre système de vote électronique vérifiable, développé par l'INRIA en France, qui vise à garantir la sécurité et la confidentialité des votes.

Expériences nationales : Le mémoire de Yousfi Izzeddine décrit plusieurs cas d'étude qui illustrent les défis et les succès de la mise en œuvre à grande échelle :

4. Estonie [11]: Ce pays a mis en place un système de vote par internet où l'authentification des électeurs se fait à l'aide d'une carte d'identité électronique et d'un code PIN, assurant l'intégrité du processus et la vérifiabilité des votes.
5. Australie (iVote) [12] : Ce système, notamment utilisé en Nouvelle-Galles du Sud, a fait l'objet d'audits de sécurité pour examiner son protocole. Des critiques ont été soulevées, mais le rapport indique qu'il n'y a pas eu d'incidents de sécurité majeurs.
6. Norvège [13] : Les essais de vote en ligne en Norvège ont démontré que le système fonctionnait correctement, mais n'ont pas fondamentalement modifié le taux de participation, ce qui a finalement conduit à l'arrêt du projet pour des raisons politiques et de coût.

1.1.3 Aspects techniques et méthodologiques de la conception

Les documents consultés mettent un accent et se complètent pour aborder les aspects techniques cruciaux de la conception d'une application de vote électronique.

Méthodologie de développement : Selon le mémoire d'Adama Seye, la méthodologie agile SCRUM est une approche pertinente pour la conception de l'application, car elle permet un développement itératif et incrémental, favorisant une meilleure adaptation aux besoins.

Modélisation et conception : Les rapports [1] et [4] soulignent l'importance du langage UML (Unified Modeling Language) pour la modélisation du système. Ce langage permet de représenter graphiquement les composants, les interactions et les flux de données, ce qui est essentiel pour la conception d'un système complexe comme une application de vote.

Technologies proposées :

Le mémoire d'Adama Seye propose des technologies spécifiques pour la mise en œuvre : un serveur web tel qu'Apache Tomcat, des langages de programmation comme Java et des bases de données comme MySQL pour le stockage des informations.

Le mémoire de Yousfi Izzeddine met l'accent sur les aspects de sécurité et de cryptographie, mentionnant l'utilisation d'algorithmes de cryptage tels que RSA et des fonctions de hachage comme SHA-256 pour garantir l'intégrité et la confidentialité des votes.

Il ressort de nos recherches que la création d'une application de vote électronique est une réponse viable aux lacunes du système de vote traditionnel. Ils soulignent que la conception d'un tel système est une démarche de génie logiciel qui doit intégrer des méthodologies rigoureuses comme SCRUM et des outils de modélisation comme UML. L'analyse des expériences internationales et des systèmes existants met en évidence l'importance de la sécurité, de la cryptographie et de l'intégrité du processus, tout en reconnaissant que l'adoption d'un tel système dépend également de facteurs sociaux et politiques.

Notons que la conception d'un système de vote a déjà aussi attirer l'attention de la communauté ulpgloise, en 2023, cette année le groupe d'étudiants GESI a conçu un système de vote électronique qui pouvait prendre en charge non seulement le vote les représentants des étudiants, mais il pouvait aussi prendre en charge le vote présidentiel et autres types de vote.

1.1.4 Tableau synthétique des systèmes de vote déjà conçu

Ce tableau dresse une étude comparative des systèmes de vote déjà conçus. Il analyse chaque solution selon ses avantages et ses inconvénients, justifiant ainsi le besoin de notre nouvelle application

Tableau 1. Tableau synthétique des systèmes de vote déjà conçu

N°	Référence	Auteurs	Types de systèmes	Technologies utiliser	Sécurité	Caractéristiques principales	Liens
1	[4]	Yousfi Izzeddine	Système de vote à distance	Vs code, css, php, mysql, html, bootstrap, xamp, CodeIgniter et uml	Accent sur la sécurité et la cryptographie.	Système de vote à distance.	Co be ut
2	[1]	Adama Seye	Système de vote électronique (Mise en œuvre)	Apache Tomcat, Java, MySQL. Méthodologie SCRUM et modélisation UML.	Vise à améliorer la transparence et l'efficacité.	Application aux élections organisées à l'UASZ. Approche par développement itératif et incrémental.	Ba tra de tra

N°	Référence	Auteurs	Types de systèmes	Technologies utiliser	Sécurité	Caractéristiques principales	Li
3	[2]	Michael J. Radwin	Protocole de vote électronique	Non spécifié	Intraçabilité (anonymat). Protection contre le double-vote. Garantit la Confidentialité et la Vérifiabilité universelle.	Technique des signatures aveugles de David Chaum. Utilisation de l'opérateur XOR.	Sa re co
4	[7]	Adida, Ben	Système de vote en ligne (Open Source)	Non spécifié	Système de vote de bout en bout vérifiable.	Utilisé par des institutions universitaires. Permet à l'électeur de vérifier son vote sans compromettre l'anonymat.	No

N°	Référence	Auteurs	Types de systèmes	Technologies utiliser	Sécurité	Caractéristiques principales	Li
5	[9]	Ralf phillipp, weinmann, janis grabis et al	Vote par Internet (i-voting)	Carte d'identité électronique et code PIN	Système d'identification numérique sécurisé, assurant l'intégrité et la vérifiabilité des votes.	Vote par Internet (i-voting) avec vérification d'identité par carte électronique et code PIN.	No
6	[11]	Non spécifié (Essais nationaux)	Essais de vote en ligne	Non spécifié dans l'extrait.	Système jugé fonctionnel.	Essais de vote au niveau national	N' tar de
7	(ULPGL)	Groupe d'étudiants GESI (2023)	Système de vote électronique	VS Code, css, php, mysql, html, bootstrap, xamp,	Fonctionnel mais rien n'est certain par rapport à la sécurité et à l'implémentation car ce système n'a jamais été	Conçu par des étudiants de l'ULPGL, ce système pouvait organiser non seulement de vote de l'ULPGL	Co él ce

N°	Référence	Auteurs	Types de systèmes	Technologies utiliser	Sécurité	Caractéristiques principales	Li
					utiliser ou soit tester même pour une élection	mais aussi on pourrait organiser de vote présidentiel et autres types d'élections.	

1.2 Généralités sur le vote électronique

1.2.1. Historique du vote électronique

Le concept du vote électronique trouve ses origines dans les années 1960, avec l'apparition des premiers **ordinateurs et systèmes automatisés de dépouillement**. À cette époque, certaines institutions ont commencé à utiliser des **machines de lecture optique** pour compter automatiquement les bulletins perforés, comme ce fut le cas aux États-Unis.

- **Années 1970 – 1980** : Les premiers prototypes de **machines à voter électroniques** apparaissent, notamment aux États-Unis et en Europe. Ces dispositifs permettaient de remplacer le bulletin papier par une interface électronique (clavier, écran ou levier) et d'enregistrer directement les votes dans une mémoire électronique. Ces machines étaient toutefois coûteuses et suscitaient déjà des débats sur la sécurité et la transparence.
- **Années 1990** : Avec la démocratisation d'Internet, l'idée du **vote en ligne (Internet voting)** commence à émerger. Certains pays et institutions testent cette méthode pour de petites élections internes (associations, universités, syndicats). Aux Pays-Bas et en Allemagne, des expérimentations sont menées, mais elles rencontrent des critiques liées à la sécurité informatique.
- **Années 2000** : Plusieurs pays passent à l'échelle nationale.
 - **Estonie (2005)** : premier pays au monde à instaurer officiellement le vote par Internet lors d'élections locales, puis nationales. Ce modèle est souvent cité comme une réussite, grâce à un système d'identification numérique sécurisé.
 - **Belgique et France** : utilisation progressive de machines à voter électroniques dans certains bureaux de vote.
 - **États-Unis** : adoption massive de machines électroniques pour remplacer le papier, surtout après les controverses de l'élection présidentielle de 2000 (affaire des bulletins perforés en Floride).
- **Années 2010 à nos jours** : Le vote électronique s'élargit à différents contextes : élections politiques, élections universitaires, élections professionnelles et associatives. Toutefois, il reste un sujet de débat : certains pays, comme l'Allemagne et les Pays-Bas, ont interdit ou

limité son usage à cause de risques de fraude informatique, tandis que d'autres (comme l'Estonie, le Brésil ou l'Inde) continuent à le déployer à grande échelle.

1.2.2. Types de votes électroniques

Le vote électronique peut se décliner sous plusieurs formes selon le dispositif et le contexte d'utilisation :

1. Le vote sur machine électronique (Electronic Voting Machine – EVM)

Les électeurs se déplacent dans un bureau de vote et expriment leur choix sur une machine dédiée (écran tactile, clavier, etc.). Les résultats sont enregistrés dans une mémoire électronique puis dépouillés automatiquement.

2. Le vote par lecture optique

L'électeur remplit un bulletin papier standardisé (cases à cocher, marques). Ce bulletin est ensuite scanné et dépouillé par une machine de lecture optique.

Cette méthode est hybride : vote papier mais dépouillement électronique.

3. Le vote en ligne (Internet Voting ou I-voting)

L'électeur vote à distance via Internet, depuis un ordinateur, une tablette ou un smartphone.

Cela offre flexibilité et gain de temps, mais pose des défis en matière de cyber sécurité.

4. Le vote par dispositifs sécurisés

L'électeur utilise une **carte à puce, une carte biométrique ou une authentification forte (empreinte digitale, reconnaissance faciale, code envoyé par SMS)** pour voter.

Cette méthode vise à renforcer la sécurité et l'identification des électeurs.

4. Le vote hybride (électronique + support papier)

Après avoir voté sur une machine électronique, une copie papier du bulletin est imprimée.

Cette trace physique sert de preuve et peut être utilisée pour un recomptage manuel.

Ce modèle est de plus en plus recommandé pour rassurer sur la transparence du processus.

1.2.3 Avantages et inconvénients des systèmes de votes électroniques

Avantages potentiels du vote électronique

- **Gain de temps et d'efficacité** : le dépouillement des votes est quasi instantané, ce qui permet d'obtenir les résultats beaucoup plus rapidement contrairement à un système de vote traditionnelle en réduisant les erreurs qui peuvent être causées par les erreurs humaines liées au comptage manuel.
- **Réduction des coûts** : à long terme, le vote électronique peut entraîner des économies significatives en réduisant les dépenses liées à l'impression et à l'acheminement des bulletins de vote
- **Meilleure accessibilité** : le vote en ligne peut faciliter la participation des électeurs qui vivent à l'étranger, des personnes infirmes ou des personnes vivant dans des zones reculées
- **Potentiel augmentation de la participation** en rendant le vote plus rapide, cela peut augmenter le taux de participation surtout pour la génération actuelle dont la jeunesse est trop accro aux outils numériques

Inconvénients et risques du vote électronique

- **Sécurité et intégrité du vote** : c'est le point critique le plus important, les systèmes de vote électronique sont des cibles potentielles pour le piratage technique et les fraudes
- **Manque de transparence** : contrairement au vote traditionnel où le dépouillement peut même se faire en public, pour le vote électronique tout se passe en arrière-plan géré par un système

- **Atteinte au secret du vote** : surtout dans le cas où le vote se passe à domicile, il est difficile de garantir que l'électeur est libre de toute influence ou de toute contrainte au moment du vote, car il n'y a pas d'isoloir physique
- **Fiabilité technique** : en cas de coupure du courant ou soit en cas de coupure de la connexion, tout le processus de vote est arrêté cela peut être pénible surtout si l'on est dans une zone où l'on n'a pas un courant stable ou soit exiger la présence de plusieurs sources d'énergie
- **Exclusion de certains électeurs** : les personnes qui n'ont pas accès à Internet auront du mal à voter et à exprimer leur choix

Malgré les différents inconvénients du système de vote électronique, nous pouvons toujours dire que le système de vote électronique reste toujours la meilleure option à adopter pour les cas d'un vote car il présente beaucoup plus d'avantages que de désavantages.

Dans le cadre de notre travail, nous allons apporter un pas de plus en concevant un système qui non seulement permettra le vote sécurisé à distance mais aussi essaiera de réduire les risques et désavantages que présente le système de vote électronique.

Conclusion partielle

Au terme de ce premier chapitre, il ressort clairement que le système de vote traditionnel, basé principalement sur l'utilisation de bulletins papier et sur un dépouillement manuel, présente de nombreuses limites tant sur le plan organisationnel que sur le plan de la transparence et de la fiabilité. Les travaux analysés dans la revue de la littérature ont mis en évidence les difficultés récurrentes liées à ce mode de vote, notamment la lenteur du processus, les risques d'erreurs humaines, la vulnérabilité aux fraudes et le faible taux de participation des électeurs.

L'étude des expériences antérieures, aussi bien académiques que nationales et internationales, a permis de montrer que le vote électronique constitue une alternative crédible pour moderniser les processus électoraux. Les systèmes existants, tels que Helios Voting, Belenios ou encore les expériences menées en Estonie, en Australie et en Norvège, démontrent que l'utilisation des

technologies de l'information peut améliorer l'accessibilité, accélérer le dépouillement et renforcer l'efficacité globale du vote. Toutefois, ces mêmes expériences soulignent également que le vote électronique soulève des défis importants, en particulier en matière de sécurité, de confidentialité, de protection des données et de respect du secret du vote.

Les généralités présentées sur le vote électronique ont permis de retracer son évolution historique, d'identifier les différents types de votes électroniques et d'analyser de manière critique leurs avantages et leurs inconvénients. Il apparaît ainsi que, malgré les risques et les limites qu'il comporte, le vote électronique demeure une solution prometteuse lorsqu'il est conçu et implémenté selon des principes rigoureux de sécurité, de transparence et de fiabilité.

En définitive, ce chapitre a posé les bases théoriques et conceptuelles nécessaires à la compréhension du problème étudié et à la justification du choix du vote électronique comme solution envisagée. Il ouvre la voie au chapitre suivant, qui portera sur l'approche conceptuelle du système proposé, en mettant l'accent sur les choix technologiques, la modélisation et les spécifications permettant de concevoir une application de vote adaptée au contexte universitaire de l'ULPGL.

Chapitre 2

APPROCHE CONCEPTUELLE DU SYSTEME DE VOTE

Ce chapitre est consacré à la modélisation technique et à la planification détaillée de l'application web de vote en ligne pour l'ULPGL. Faisant suite à la revue de la littérature et à l'analyse des lacunes du système traditionnel, ce chapitre établit les choix architecturaux et les exigences précises permettant de transformer les besoins identifiés en une solution logicielle concrète.

Dans un premier temps, nous débuterons par la présentation des technologies à utiliser, où sera justifiée l'adoption d'une architecture client-serveur. Dans un second temps, la présentation des tableaux de spécifications détaillant l'ensemble des exigences du système. Nous distinguerons clairement les spécifications fonctionnelles des spécifications non fonctionnelles.

Ce chapitre sert ainsi de pont essentiel entre l'analyse théorique et l'implémentation pratique, jetant les bases structurelles de la solution proposée.

2.1 Analyse du système de vote

L'analyse du système de vote au sein de l'ULPGL a permis de mettre en évidence des besoins critiques en termes de transparence, de rapidité et de sécurité. Pour répondre à ces défis, le passage d'un système manuel à une solution numérique automatisée impose l'utilisation de technologies robustes et éprouvées.

2.1.1 Analyses des besoins et contraintes du système

L'analyse du processus électoral actuel révèle plusieurs points de friction qui dictent les choix techniques futurs :

- **Accessibilité et mobilité** : le système doit être accessible depuis n'importe quel terminal pour permettre aux plus de 3000 étudiants de voter sans contrainte géographique

- **Intégrité et sécurité** : la protection contre la fraude et les cyberattaques est impérative. Cela nécessite des mécanismes de chiffrement et d'authentification forts.
- **Performance et disponibilité** : le système doit supporter des connexions simultanées massives lors des pics de vote sans interruption de service.
- **Transparence des résultats** : le dépouillement doit être automatique et les résultats proclamés en temps réel pour garantir la crédibilité du scrutin.

2.2 Présentation des technologies à utiliser

Pour concrétiser notre projet d'application de vote en ligne, nous nous sommes appuyés sur des technologies web standards et éprouvées. Nous avons choisi les outils en nous orientant vers une architecture client-serveur où les tâches seront réparties entre le navigateur de l'utilisateur (le front-end) et le serveur web (le back-end). Avec cette approche, la robustesse, la sécurité et la scalabilité nécessaires pour un système de vote seront assurées.

2.2.1 Technologies Front-end : L'interface utilisateur

Les technologies front-end sont la face visible de l'application. Elles gèrent toute la partie visible avec laquelle les acteurs interagiront directement dans leur navigateur.

2.2.1.1 HTML (HyperText Markup Language)

Le HTML [14] est le langage de balisage qui sert de structure de base à toute page web. Il ne s'agit pas d'un langage de programmation, mais du squelette de l'application. Son rôle est de définir et d'organiser les éléments qui constituent l'interface, comme les titres, les formulaires de connexion, les listes de candidats, et les boutons de vote. Dans le cadre de ce projet, le HTML est la fondation sur laquelle toutes les autres technologies s'appuient pour présenter le contenu de manière sémantique et structurée.



Figure 1 logo html

2.2.1.2 CSS (Cascading Style Sheets)

Le CSS [15] est responsable de la mise en forme et de l'apparence visuelle des éléments HTML. Il permet de transformer le squelette de l'application en une interface attrayante et intuitive. Le CSS est utilisé pour définir le style, les couleurs, les polices de caractères, ainsi que la mise en page de



Figure 2 logo css

l'ensemble de l'application. Il joue un rôle crucial dans l'amélioration de l'expérience utilisateur (UX) en garantissant que l'interface est claire, lisible et cohérente sur différents appareils (ordinateurs, tablettes, téléphones).

2.2.1.3 JavaScript

Le JavaScript [16] est un langage de programmation qui apporte l'interactivité et le dynamisme au front-end. Il s'exécute directement dans le navigateur de l'utilisateur. Il permet de gérer les actions de l'utilisateur en temps réel, comme la validation des formulaires avant leur envoi, l'affichage de messages de confirmation de vote sans recharger la page, ou la création d'animations et d'effets visuels pour une expérience plus fluide. Il est le "cerveau" qui rend l'interface vivante et réactive.



Figure 3 logo javascript

2.2.2 Technologies Back-end : La logique et la sécurité

Les technologies back-end s'exécutent côté serveur et sont essentielles pour la gestion de la logique métier, l'authentification des utilisateurs, et la sécurité des données de vote.

2.2.2.1 PHP (Hypertext Preprocessor)

Le PHP [17] est un langage de script côté serveur. Il est le moteur de l'application. Son rôle principal est de traiter les requêtes envoyées par le front-end. Par exemple, lorsqu'un utilisateur se connecte ou vote, le PHP interprète cette requête, exécute la logique de vérification et d'enregistrement, puis renvoie une réponse au navigateur. Le PHP est fondamental pour la sécurité des votes, l'authentification des utilisateurs, et la gestion de toutes les interactions avec la base de données.

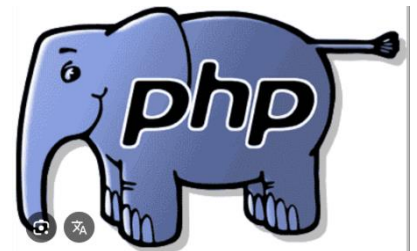


Figure 4 logo php

2.2.2.2 SQL (Structured Query Language)

Le SQL [18] est un langage standard utilisé pour interagir avec les bases de données relationnelles. C'est la manière dont l'application stocke, organise et récupère toutes les informations essentielles. Il sera utilisé pour :



Figure 5 logo sql

- Stocker les données des étudiants, des candidats, et des votes.

- Gérer l'intégrité des données pour s'assurer que chaque étudiant ne vote qu'une seule fois.
- Interroger la base de données pour générer les résultats finaux de l'élection.

C'est avec l'utilisation de ces technologies citées ci-haut que nous comptons réaliser notre projet de création d'une application web de vote.

2.2.3. Le langage de modélisation

Au-delà de ces technologies citées ci-haut, nous allons utiliser aussi le langage UML (Unified Modeling Language) [19] qui est en effet un langage de modélisation graphique en développement logiciel et en conception orientée objet. Pour la représentation graphique de notre système, nous allons utiliser l'application draw.io qui est une application qui utilise le langage UML et qui va nous servir pour la représentation des différents diagrammes de notre système, entre autres les diagrammes de classe, diagramme de séquence, diagrammes d'activités et autres diagrammes



Figure 6 logo UML

2.3 Cahier des charges détaillé du projet

Ce cahier des charges sert de document de référence pour le développement technique de l'application.

2.3.1 Objectifs du système

L'objectif principal est de concevoir une application web permettant d'organiser les élections du Porte-parole, des Présidents facultaires et des Chefs de promotion de l'ULPGL de manière fiable, rapide et sécurisée.

2.3.2 Spécifications fonctionnelles (Besoins métiers)

Le système doit impérativement couvrir les modules suivants :

Module Électeur : Authentification unique, consultation de la liste des candidats par catégorie, expression du vote et confirmation de la prise en compte du suffrage.

Module Candidat : Affichage du profil et des ambitions électorales au sein de l'interface de vote.

Module Administration : Gestion de la liste électorale (importation des étudiants), configuration des dates de scrutin, supervision en temps réel et proclamation des résultats finaux.

2.3.3 Exigences de sécurité et de fiabilité

Anonymat : Le système doit dissocier l'identité de l'électeur de son bulletin de vote dans la base de données pour garantir le secret du suffrage.

Authentification forte : Accès restreint aux seuls étudiants inscrits via des identifiants universitaires vérifiés.

Protection des données : Utilisation de requêtes préparées pour contrer les injections SQL et sécuriser les échanges entre le client et le serveur.

2.3.4. Contraintes de performance

Charge simultanée : Le système doit pouvoir gérer des pics de connexions simultanées (jusqu'à 3000 étudiants potentiels) sans dégradation du temps de réponse.

Disponibilité : L'application doit rester opérationnelle durant toute la période définie pour le scrutin.

2.3.5. Méthodologie de développement

Nous adoptons la méthodologie Agile, permettant un développement itératif. Cela nous offre la flexibilité nécessaire pour adapter le système aux retours des utilisateurs (étudiants et administration) durant les phases de test.

Cette illustration ci bas schématise le processus de vote en ligne, elle montre l'interaction fluide entre l'électeur et l'interface numérique, garantissant l'accessibilité et la simplicité du parcours utilisateur.

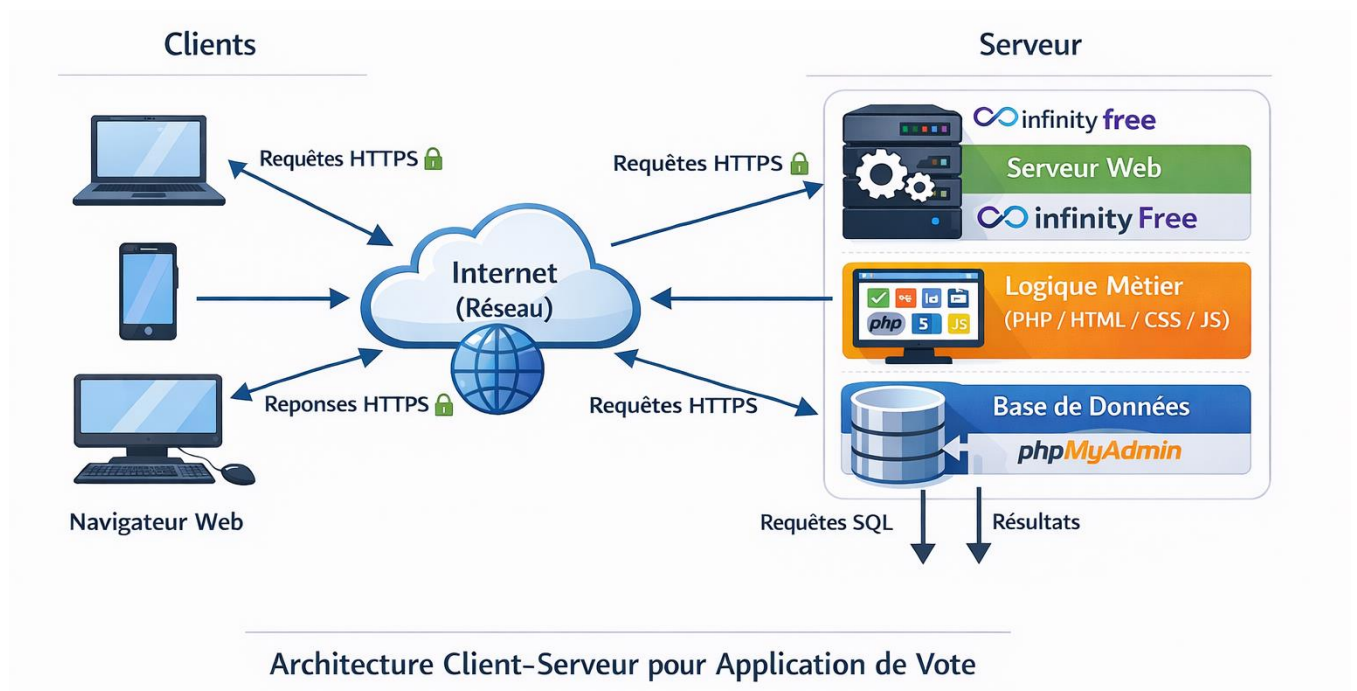


Figure 7 architecture client-serveur pour application web de vote

2.3 Présentation du tableau des spécifications du système

2.3.1 Tableau de spécifications fonctionnelles

Le tableau suivant présente les spécifications fonctionnelles de notre application de vote, structurées de manière à détailler précisément chaque action attendue du système. Il se décompose en plusieurs colonnes clés permettant d'identifier chaque besoin. L'ID pour le référencement unique de la fonction, la spécification pour décrire l'action ou la fonctionnalité à réaliser, et l'état, l'effort, la stabilité et la cible qui donne une description précisant les règles métier et le comportement attendu du système pour répondre aux défis de transparence et de sécurité identifiés.

Tableau 2 Tableau de spécifications fonctionnelles

ID	Spécification	Etat	Criticisme	Effort	Stabilité	Cible
1	Le système doit permettre aux étudiants se trouvant dans la base de	Approuvé	Critique	5h	Stable	1.0

	données de s'inscrire en insérant le nom, le post nom, le matricule et l'adresse mail, puis se connecter en insérant son adresse mail et le mot de passe qui sera généré par le système et envoyer à l'adresse mail fournie					
2	Le système doit permettre l'importation de la liste d'étudiant à partir d'un fichier csv comportant le prénom, le nom, le matricule, l'adresse mail, la date de naissance, l'adresse, la faculté et la promotion,	Planifié	Moyen	2h	Stable	
3	Le système doit permettre à un utilisateur de soumettre sa candidature pour une élection en cours de préparation. La soumission doit inclure des informations telles que le nom du candidat, une photo, un CV et une lettre de motivation	Planifié	Critique	3h	Stable	1.0
4	Le système doit permettre à un candidat de suivre le statut de sa candidature (en attente de validation, approuvée, rejetée).	Planifié	Moyen	1h	Stable	1.0
5	Le système doit permettre à un administrateur d'examiner et de valider ou de rejeter les candidatures soumises.	Planifié	Moyen	2h	Stable	1.0
6	Le système doit permettre à un administrateur de créer un nouveau vote en renseignant le titre, une	Planifié	Critique	5h	Stable	1.0

	description, le type de vote, les dates de début et de fin de vote,					
7	Le système doit permettre à l'administrateur d'attribuer les candidats validés à une élection spécifique et de définir leur ordre d'apparition sur le bulletin de vote.	Planifié	Critique	3h	Stable	1.0
8	Le système doit permettre à un électeur authentifié d'accéder à une liste des élections pour lesquelles il est éligible.	Planifié	Moyen	2h	Stable	1.0
9	Le système doit afficher un message de confirmation à l'électeur une fois son vote soumis et enregistré avec succès	Planifié	Moyen	1h	Stable	1.0
10	Le système doit calculer automatiquement les résultats et les afficher sous forme de graphiques et de tableaux récapitulatifs.	Planifié	Critique	4h	Stable	1.0
11	Le système doit permettre à l'administrateur de générer des rapports exportables sur la participation, les résultats, et la diffusion des résultats d'un vote en état termine par mail à tous les participants de ce vote.	Planifié	Moyen	4h	Stable	1.0

2.3.2 Tableau des spécifications non fonctionnelles

Ce tableau présente les spécifications non fonctionnelles qui définissent les exigences de qualité et les contraintes de performances du système. Il s'articule autour de sept colonnes essentielles qui décrivent de manière assez détaillée les seuils techniques à respecter pour garantir une expérience utilisateur optimale et sécurisée.

Tableau 3 Tableau des spécifications non fonctionnelles

ID	Spécification	Etat	Criticisme	Effort	Stabilité	Cible
1	Le système doit garantir que le lien entre l'électeur et son vote est indéchiffrable. Le vote doit être anonymisé et stocké séparément de l'identité de l'électeur.	Planifié	Critique	4h	Stable	1.0
2	Le système doit être protégé contre les attaques par injection SQL, XSS, et les attaques par déni de service.	Planifié	Critique	7h	Stable	1.0
3	En cas de défaillance matérielle ou logicielle, le système doit pouvoir récupérer l'état des données (votes enregistrés) sans perte et dans un délai minimal.	Planifié	Critique	4h	Stable	1.0
4	Le temps de chargement de la page de vote et de soumission du vote ne doit pas excéder un seuil, même en cas de forte affluence.	Planifié	Moyen	1h	Stable	1.0
5	Le système doit pouvoir supporter un nombre élevé de connexions simultanées sans dégradation significative des performances.	Planifié	Critique	5h	Stable	1.0

6	Le service de vote en ligne doit être accessible 24h/24 et 7j/7 pendant la période de vote.	Planifié	Moyen	4h	Stable	1.0
7	Le calcul des résultats doit être 100% exact et reproductible. Toute erreur de calcul est inacceptable	Planifié	Critique	6h	Stable	1.0
8	Le système doit respecter les lois sur la protection des données	Planifié	Critique	4h	Stable	1.0
9	L'interface doit être simple, intuitive et adapter à tous les écrans possibles	Planifié	Critique	3h	Stable	1.0
10	Le système ne permet pas l'annulation ou la modification du vote une fois qu'il est soumis	Planifié	Critique	3h	Stable	1.0
11	Le système doit envoyer des alertes aux administrateurs en cas de tentatives d'accès non autorisées ou d'erreurs techniques critiques.	Planifié	Moyen	5h	Stable	1.0
12	Le système doit garantir qu'un électeur ne peut voter qu'une seule fois par scrutin. Après avoir voté, son statut doit être mis à jour comme "A voté".	Planifié	Moyen	5h	Stable	1.0

2.4 Présentation des différents diagrammes de modélisation

L'approche conceptuelle de notre système constitue une étape fondamentale dans le processus de conception globale de l'application. Elle se matérialise principalement par la phase de modélisation, dont le rôle est de traduire de manière rigoureuse et structurée les besoins fonctionnels et non fonctionnels exprimés lors de l'analyse, en une architecture logicielle cohérente, compréhensible et vérifiable.

Dans cette partie nous faisons recours au Langage de Modélisation Unifié (UML), qui représente un standard reconnu dans le domaine de l'ingénierie logicielle et de la conception orientée objet. UML offre un ensemble de notations graphiques permettant de décrire, de façon abstraite mais précise, les différents aspects d'un système informatique, tant sur le plan structurel que comportemental. Son utilisation favorise une meilleure communication entre les concepteurs, les développeurs et les parties prenantes, tout en réduisant les ambiguïtés liées à l'interprétation des exigences du système.

L'objectif principal de cette phase de modélisation est de visualiser clairement l'architecture du système de vote en ligne, en mettant en évidence :

- La structure statique de l'application, notamment les entités, leurs attributs et les relations existantes entre elles ;
- La dynamique comportementale du système, c'est-à-dire les interactions entre les acteurs et le système, ainsi que les échanges de messages entre les différents composants lors de l'exécution des fonctionnalités clés, telles que l'authentification, le vote et la gestion des résultats.

Les différents diagrammes UML présentés dans cette section, notamment les diagrammes de cas d'utilisation, de séquence et de classes, ont été conçus à l'aide de l'outil draw.io, choisi pour sa simplicité, sa flexibilité et sa capacité à produire des schémas clairs et lisibles. Ces diagrammes constituent un support essentiel à la compréhension du fonctionnement global du système, en illustrant de manière visuelle les processus internes et les interactions entre les acteurs.

Enfin, cette modélisation joue un rôle crucial dans la phase d'implémentation qui sera abordée au chapitre suivant. Elle sert de référence technique pour le développement de l'application, en garantissant que les choix de conception respectent les exigences initiales du projet. Grâce à cette approche méthodique, le système de vote en ligne pourra répondre aux critères de fiabilité, de rapidité d'exécution et de sécurité, indispensables pour assurer l'intégrité du processus électoral et la confiance des utilisateurs.

2.4.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation constitue le point de départ fondamental de la modélisation UML du système de vote électronique. Il permet de représenter, de manière globale et synthétique, les

fonctionnalités offertes par le système ainsi que les interactions entre les acteurs externes et l'application. Ce diagramme fournit une vue fonctionnelle du système, indépendante des choix techniques d'implémentation, ce qui en fait un outil essentiel pour la compréhension et la validation des besoins exprimés.

Dans le cadre de notre application de vote en ligne, le diagramme de cas d'utilisation donné à la figure 8 vise principalement à délimiter le périmètre fonctionnel du système, en identifiant clairement ce que le système doit faire et pour qui il le fait. Il met en évidence les différents rôles impliqués dans le processus de vote tels que l'électeur ou l'administrateur et les services auxquels chacun de ces acteurs peut accéder. Cette représentation permet ainsi d'éviter toute ambiguïté quant aux responsabilités et aux droits associés à chaque acteur.

Avant d'aboutir à la conception du diagramme de cas d'utilisation, plusieurs préalables méthodologiques ont été nécessaires. En premier lieu, une analyse approfondie des besoins fonctionnels a été réalisée à partir des exigences recueillies auprès des utilisateurs potentiels du système, notamment les étudiants, les candidats et l'administration universitaire. Cette analyse a permis d'identifier les principales fonctionnalités attendues, telles que l'authentification des électeurs, la consultation des scrutins disponibles, la soumission du vote, la gestion des candidatures et le calcul des résultats.

En second lieu, les acteurs du système ont été identifiés et définis en fonction de leur interaction directe ou indirecte avec l'application. Chaque acteur représente une entité externe au système qui déclenche ou reçoit des services. Cette étape est cruciale, car elle conditionne la cohérence du diagramme et garantit que toutes les parties prenantes sont correctement prises en compte.

Par ailleurs, une délimitation claire de la frontière du système a été établie afin de distinguer les fonctionnalités relevant du système de vote électronique de celles qui sont externes ou hors périmètre. Cette frontière permet de préciser ce qui est sous la responsabilité directe de l'application et ce qui relève d'autres systèmes ou processus organisationnels.

Une fois ces préalables définis, les différents cas d'utilisation ont été formalisés sous forme d'actions ou de services rendus par le système à chaque acteur. Chaque cas d'utilisation correspond à un objectif précis poursuivi par un acteur et décrit un ensemble cohérent d'interactions menant à un résultat observable et mesurable. L'ensemble de ces cas d'utilisation constitue une traduction fidèle des spécifications fonctionnelles du système.

Ainsi, le diagramme de cas d'utilisation joue un rôle central dans la conception de notre application de vote en ligne. Il sert non seulement de support de communication entre les concepteurs et les utilisateurs, mais également de référence structurante pour les étapes ultérieures de modélisation détaillée et d'implémentation. En offrant une vision claire et exhaustive des fonctionnalités du système, il contribue à assurer la cohérence globale du projet et la conformité de l'application aux besoins réels du contexte universitaire.

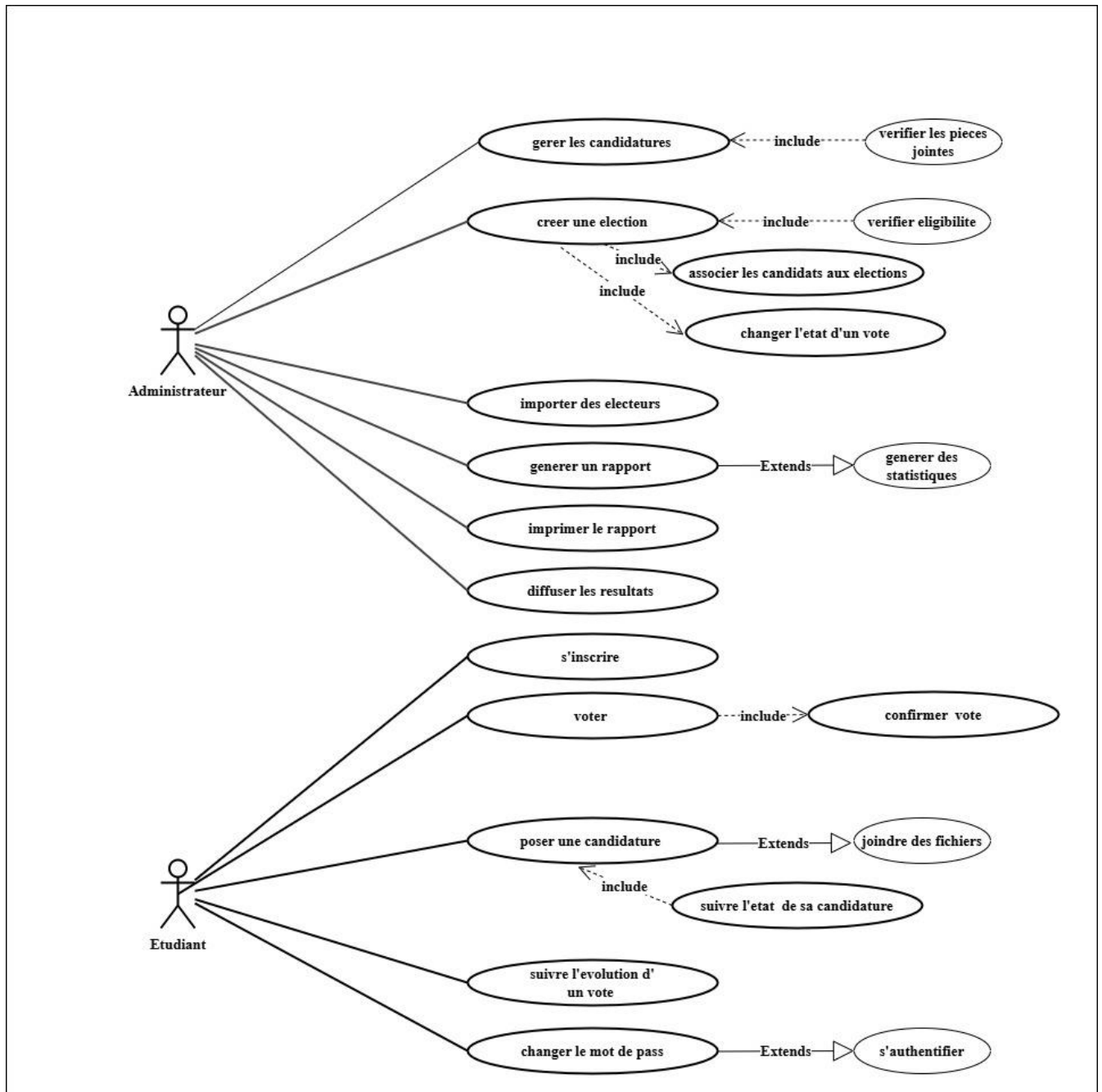


Figure 8 diagramme de cas d'utilisation

2.4.1.1 Documentation des cas d'utilisation

Le tableau 4 suivant présente les spécifications détaillées du cas d'utilisation relatif à l'inscription et à l'authentification des étudiants sur le système, elle parcourt toutes les détails par rapport à l'inscription, et la connexion au système i-voting-ulppl.

Tableau 4 documentation du CU inscription et connexion

ID	1
Description brève	Le système permet à un étudiant de s'inscrire et de se connecter.
Acteurs primaires	Étudiant
Acteurs secondaires	Système
Préconditions	Étudiant enregistré, système disponible
Enchaînement principal	Saisie informations → connexion Génération mot de passe → inscription
Postconditions	Étudiant authentifié
Enchaînements alternatifs	Informations invalides

Ce tableau numéro 5 détaille la spécification du processus d'importation massive des électeurs, cela permet d'intégrer rapidement la liste officielle des étudiants éligibles au vote.

Tableau 5 documentation du CU importation de la liste de étudiants

ID	2
Description brève	Importation de la liste officielle des étudiants.
Acteurs primaires	Administrateur
Acteurs secondaires	Base de données universitaire
Préconditions	Administrateur authentifié
Enchaînement principal	Lancement import → Enregistrement
Postconditions	Liste disponible
Enchaînements alternatifs	Échec de connexion

Ce tableau numéro 6 décrit le flux d'événements permettant aux candidats de s'enregistrer et de postuler à une élection.

Tableau 6 documentation du CU Soumission d'une candidature

ID	3
Description brève	Soumission d'une candidature à une élection.
Acteurs primaires	Candidat
Acteurs secondaires	Système
Préconditions	Utilisateur authentifié
Enchaînement principal	Remplir formulaire → Soumettre
Postconditions	Candidature enregistrée
Enchaînements alternatifs	Dossier incomplet

Ce tableau numéro 7 expose les étapes permettant à un postulant de suivre en temps réel l'évolution de son dossier, de la soumission à la validation finale par la commission électorale.

Tableau 7 documentation du CU suivi du statut de la candidature

ID	4
Description brève	Consultation de l'état d'une candidature.
Acteurs primaires	Candidat
Acteurs secondaires	Système
Préconditions	Candidat authentifié
Enchaînement principal	Accès espace → Affichage statut
Postconditions	Statut consulté
Enchaînements alternatifs	Aucun

Ce tableau numéro 8 décrit les mécanismes de contrôle permettant aux autorités compétentes entre autre l'admin de valider ou de rejeter officiellement une candidature après avoir été soumis par un des étudiants.

Tableau 8 documentation du CU validation ou rejet d'une candidature

ID	5
Description brève	Validation ou rejet d'une candidature.
Acteurs primaires	Administrateur
Acteurs secondaires	Système
Préconditions	Administrateur authentifié
Enchaînement principal	Consulter → Valider/Rejeter
Postconditions	Statut mis à jour
Enchaînements alternatifs	Aucun

Ce tableau numéro 9 détaille les paramètres nécessaires à l'initialisation d'un nouveau scrutin, incluant ainsi le titre de l'élection, la date de début et de fin et autres informations nécessaires.

Tableau 9 documentation du CU création d'un vote

ID	6
Description brève	Création d'une nouvelle élection.
Acteurs primaires	Administrateur
Acteurs secondaires	Système
Préconditions	Administrateur authentifié
Enchaînement principal	Saisie infos → Enregistrement
Postconditions	Vote créé
Enchaînements alternatifs	Infos manquantes

Ce tableau numéro 10 décrit les procédures permettant d'associer officiellement les candidats validés aux scrutins correspondants, garantissant ainsi que chaque nom apparaisse sur le bon bulletin numérique.

Tableau 10 documentation du CU attribution des candidats a une élection

ID	7
Description brève	Association des candidats validés à une élection.

Acteurs primaires	Administrateur
Acteurs secondaires	Système
Préconditions	Candidats validés
Enchaînement principal	Sélection → Attribution
Postconditions	Candidats assignés
Enchaînements alternatifs	Aucun

Ce tableau numéro 11 détaille les étapes permettant à l'étudiant authentifié d'accéder à la liste des différents scrutins auxquels il est autorisé à participer.

Tableau 11 documentation du CU consultation des élections disponibles

ID	8
Description brève	Consultation des élections disponibles.
Acteurs primaires	Électeur
Acteurs secondaires	Système
Préconditions	Électeur authentifié
Enchaînement principal	Accès liste → Affichage
Postconditions	Élections visibles
Enchaînements alternatifs	Aucun

Ce tableau numéro 12 décrit le processus technique du vote proprement dit, de la sélection du candidat par l'électeur jusqu'à l'enregistrement sécurisé du choix dans la base de données

Tableau 12 documentation du CU voter

ID	9
Description brève	Garantie d'un seul vote par électeur.
Acteurs primaires	Électeur
Acteurs secondaires	Système
Préconditions	Électeur authentifié
Enchaînement principal	Choix candidat → Vérification → Enregistrement
Postconditions	Vote enregistré

Enchaînements alternatifs	Tentative de double vote
---------------------------	--------------------------

Ce tableau numéro 13, présente l'étape cruciale de validation finale, ou le système demande à l'électeur de confirmer son choix avant l'enregistrement définitif et irréversible du bulletin dans le système.

Tableau 13 documentation du CU confirmer vote

ID	10
Description brève	Confirmation de l'enregistrement du vote.
Acteurs primaires	Électeur
Acteurs secondaires	Système
Préconditions	Vote soumis
Enchaînement principal	Enregistrement → Message
Postconditions	Vote confirmé
Enchaînements alternatifs	Erreur système

Ce tableau numéro 14 détaille le processus de dépouillement automatique, permettant une agrégation instantanée et précise des voix dès la clôture du scrutin

Tableau 14 documentation du CU calcul et affichage des résultats

ID	11
Description brève	Calcul automatique et affichage des résultats.
Acteurs primaires	Administrateur
Acteurs secondaires	Système
Préconditions	Vote clôturé ou en cours
Enchaînement principal	Calcul → Affichage
Postconditions	Résultats disponibles
Enchaînements alternatifs	Aucun

Ce tableau numéro 15 ci-dessous détaille les fonctionnalités d'exportation de résultats et de diffusion des résultats à tous les participants du vote, entre autre les candidats et les électeurs.

Tableau 15 documentation du CU générer et diffuser résultats

ID	12
Description brève	Génération et diffusion des rapports de vote.
Acteurs primaires	Administrateur
Acteurs secondaires	Système
Préconditions	Vote terminé
Enchaînement principal	Génération → Envoi
Postconditions	Rapports diffusés
Enchaînements alternatifs	Échec d'envoi

2.4.2 Diagrammes de séquence du processus de vote

Le diagramme de séquence modélise les interactions dynamiques et l'ordre chronologique des messages entre les objets du système. Il est particulièrement pertinent pour les processus critiques de la sécurité,

2.4.2.1 Authentification (SF01, SF02)

Le diagramme donné à la figure 9 illustre le processus de sécurisation de l'accès à la plateforme. L'interaction commence par la saisie des identifiants par l'utilisateur. Le système vérifie la présence de l'utilisateur en base de données. Ce mécanisme garantit que même en cas d'accès non autorisé à la base de données, les mots de passe restent illisibles. La séquence se termine soit par la création d'une session utilisateur, soit par un message d'erreur explicite en cas d'échec.

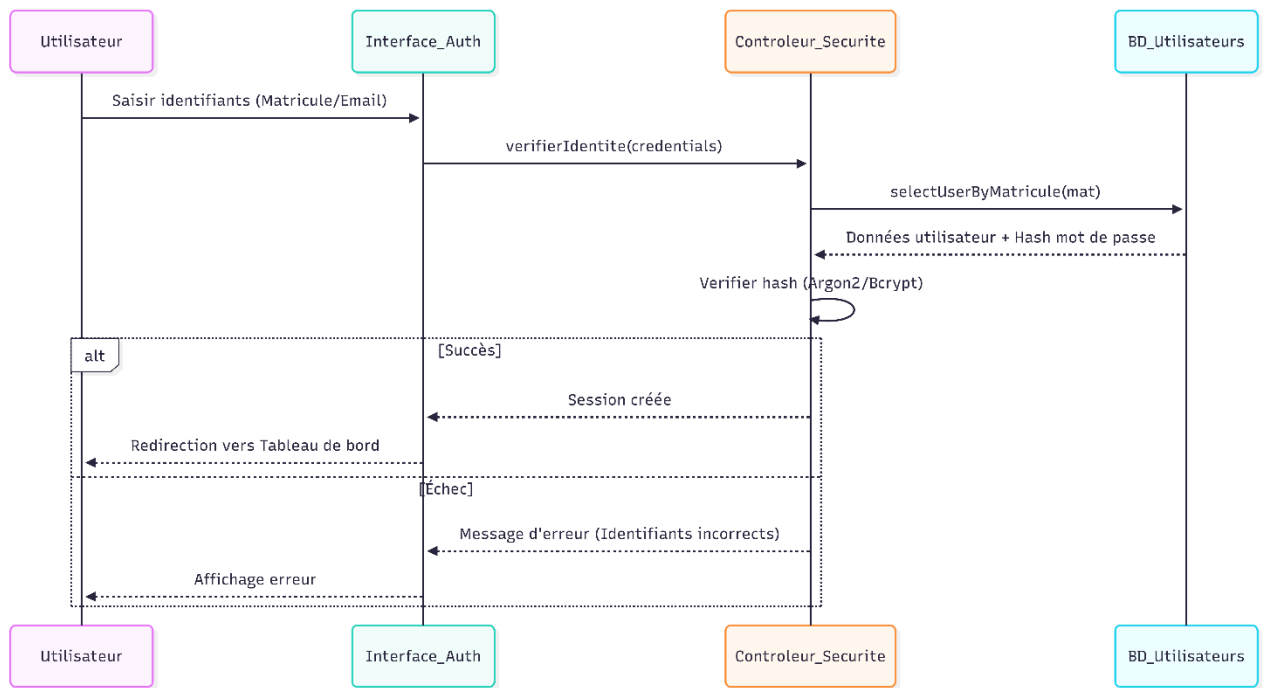


Figure 9 diagramme de la séquence inscription

2.4.2.2 Soumission de Candidature (SF04, SF05)

La gestion des candidatures est un processus à deux étapes représentées à la figure 10. Dans un premier temps, le candidat transmet ses données numériques et ses fichiers justificatifs (CV, photos). Le système sépare le stockage : les fichiers lourds vont dans un serveur de stockage dédié tandis que les références (liens) et les informations textuelles sont enregistrées en base de données avec un statut "en attente". Dans un second temps, le diagramme montre l'intervention de l'administrateur qui, après examen, met à jour le statut, déclenchant ainsi l'éligibilité du candidat pour le scrutin.

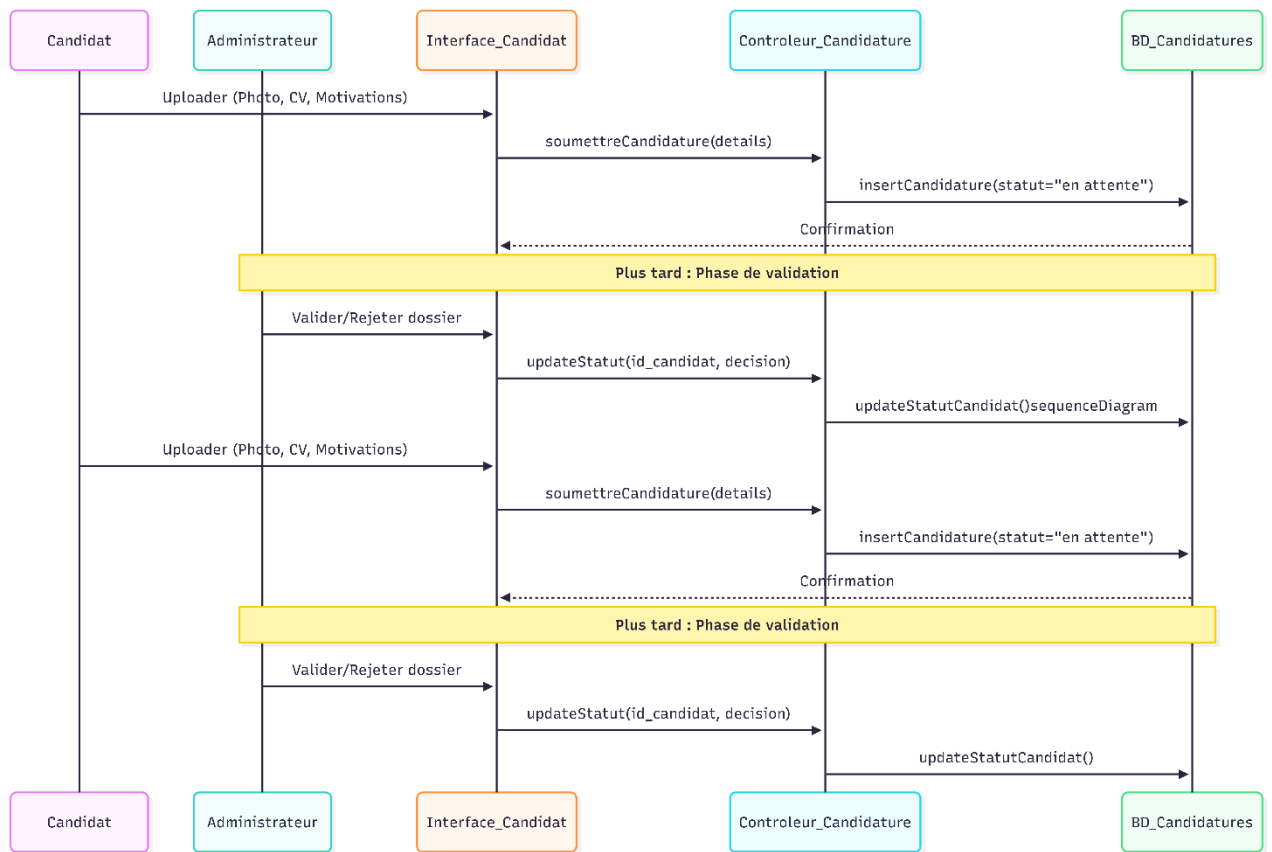


Figure 10 diagramme de la séquence soumission de candidature

2.4.2.3 Configuration d'un Scrutin (SF06)

Ce diagramme décrit à la figure 11 présente la logique métier de l'administrateur lors de la création d'une élection. L'accent est mis sur la validation des paramètres : le contrôleur d'élection vérifie la cohérence chronologique (la date de fin doit être postérieure à la date de début) avant toute insertion en base de données. Une fois validé, le scrutin est programmé et devient visible pour les électeurs concernés selon les critères de faculté ou de promotion définis.

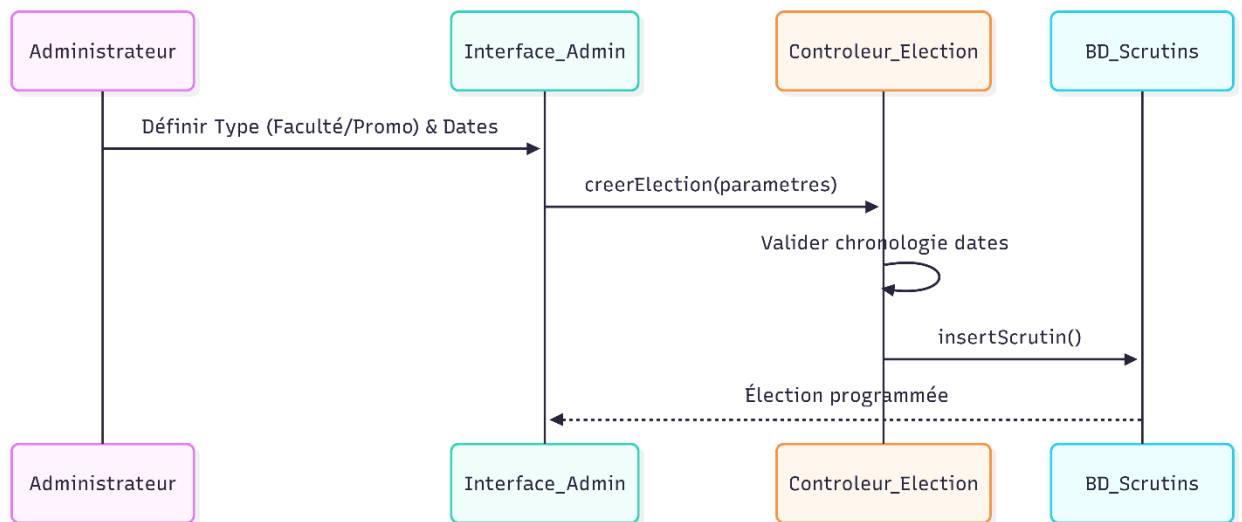


Figure 11 diagramme de la séquence création élection

2.4.2.4 Processus de Vote Sécurisé (SF09, SF10)

Ce diagramme décrit à la figure 12 est le plus critique concernant la fiabilité du système. Il met en évidence le mécanisme d'anonymisation du suffrage. Pour garantir l'intégrité du vote, le système effectue une double opération atomique : il incrémente le compteur de voix du candidat choisi et, simultanément, marque l'électeur comme ayant voté dans une table distincte. Cette séparation logique permet de garantir qu'on ne peut pas lier un bulletin à l'identité d'un étudiant, tout en rendant le double vote techniquement impossible.

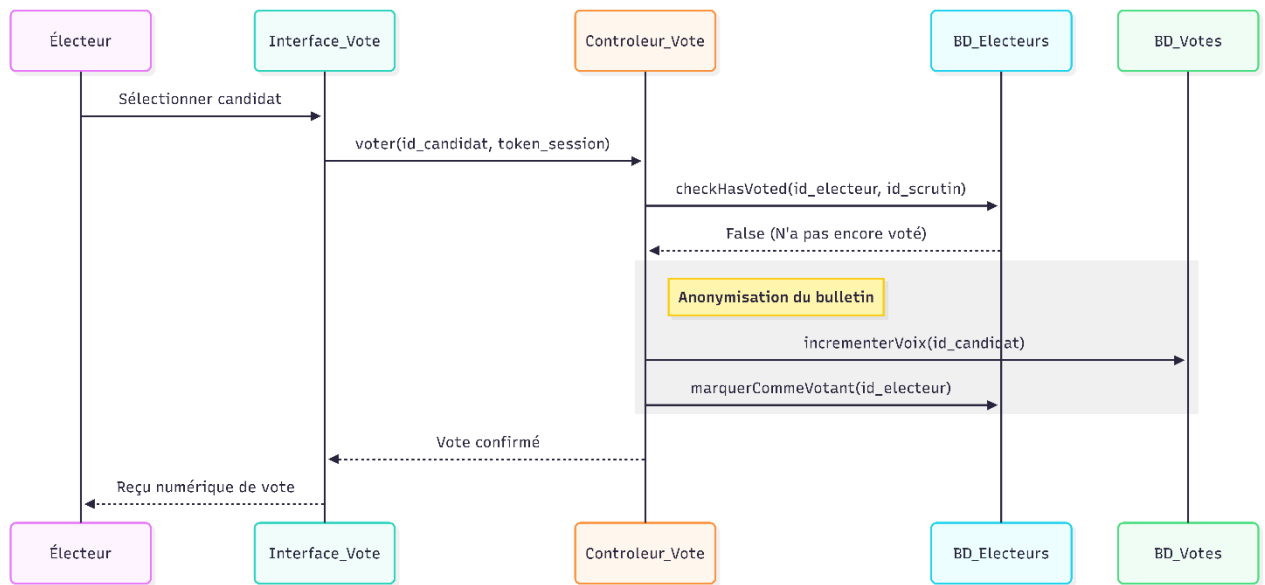


Figure 12 diagramme de la séquence vote

2.4.2.5 Proclamation des Résultats (SF11, SF12)

Ce dernier diagramme décrit à la figure 13 présente l'automatisation de la fin d'un scrutin. À l'échéance de la date limite, le système déclenche un service de calcul qui agrège les voix. Contrairement à un dépouillement manuel, le contrôleur calcule instantanément les pourcentages et le taux de participation. Les résultats sont ensuite poussés vers l'interface publique pour une transparence totale, tout en permettant à l'administrateur d'exporter les rapports officiels.

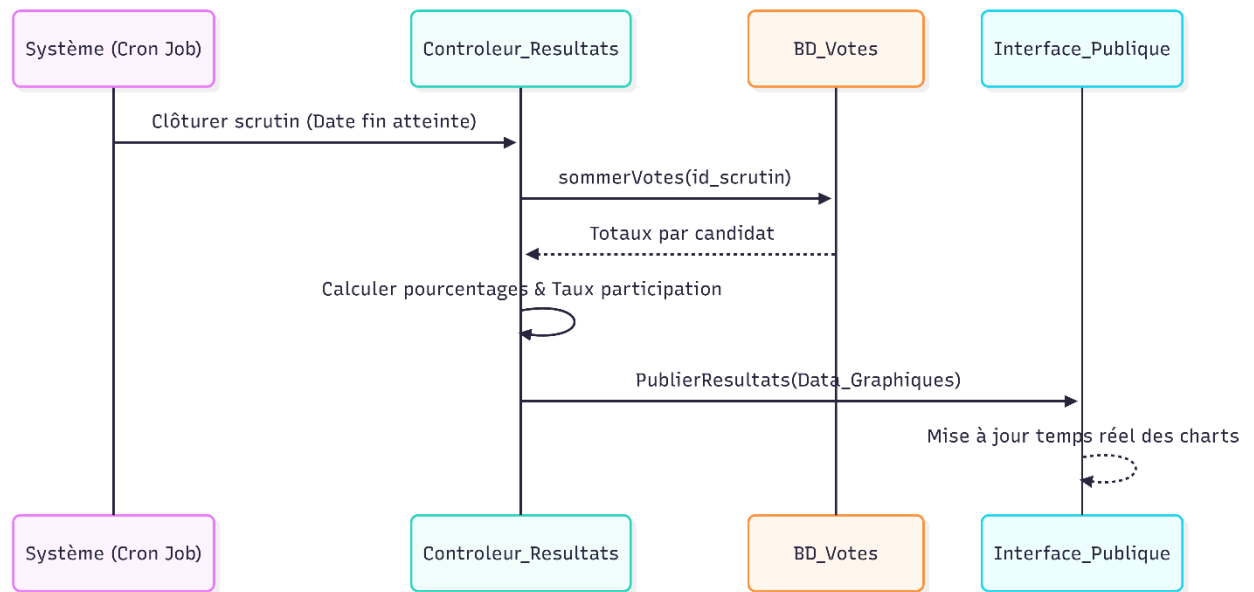


Figure 13 diagramme de la séquence proclamation des résultats

2.4.3 Diagramme de classe

Le diagramme de classes se trouvant à la figure 14 présente la structure de données statique de l'application. Il met en évidence les entités principales issues de l'analyse des besoins et leurs interdépendances :

- Entité Voter (Électeur) : C'est le socle du système. Chaque étudiant possède un matricule unique et des identifiants d'accès. Un électeur peut devenir un Candidate (relation 1 à 0...1).
- Entité Election (Scrutin) : Elle définit le cadre du vote (type, dates, faculté). Elle est liée à la table Candidacy qui fait le pont entre un candidat et une élection spécifique.
- Entité Vote : Elle enregistre le suffrage. Conformément aux contraintes de sécurité, elle est liée à l'élection et à la candidature, mais elle n'est pas directement liée à l'électeur pour garantir l'anonymat.
- Entité VoterToken : Cette table est cruciale pour la sécurité. Elle permet de générer un jeton unique par élection pour chaque électeur, garantissant qu'un étudiant ne vote qu'une seule fois (contrainte d'unicité sur `voter_id` et `election_id`).
- Entité AdminCode : Table indépendante permettant la gestion des accès administratifs pour la configuration du système.

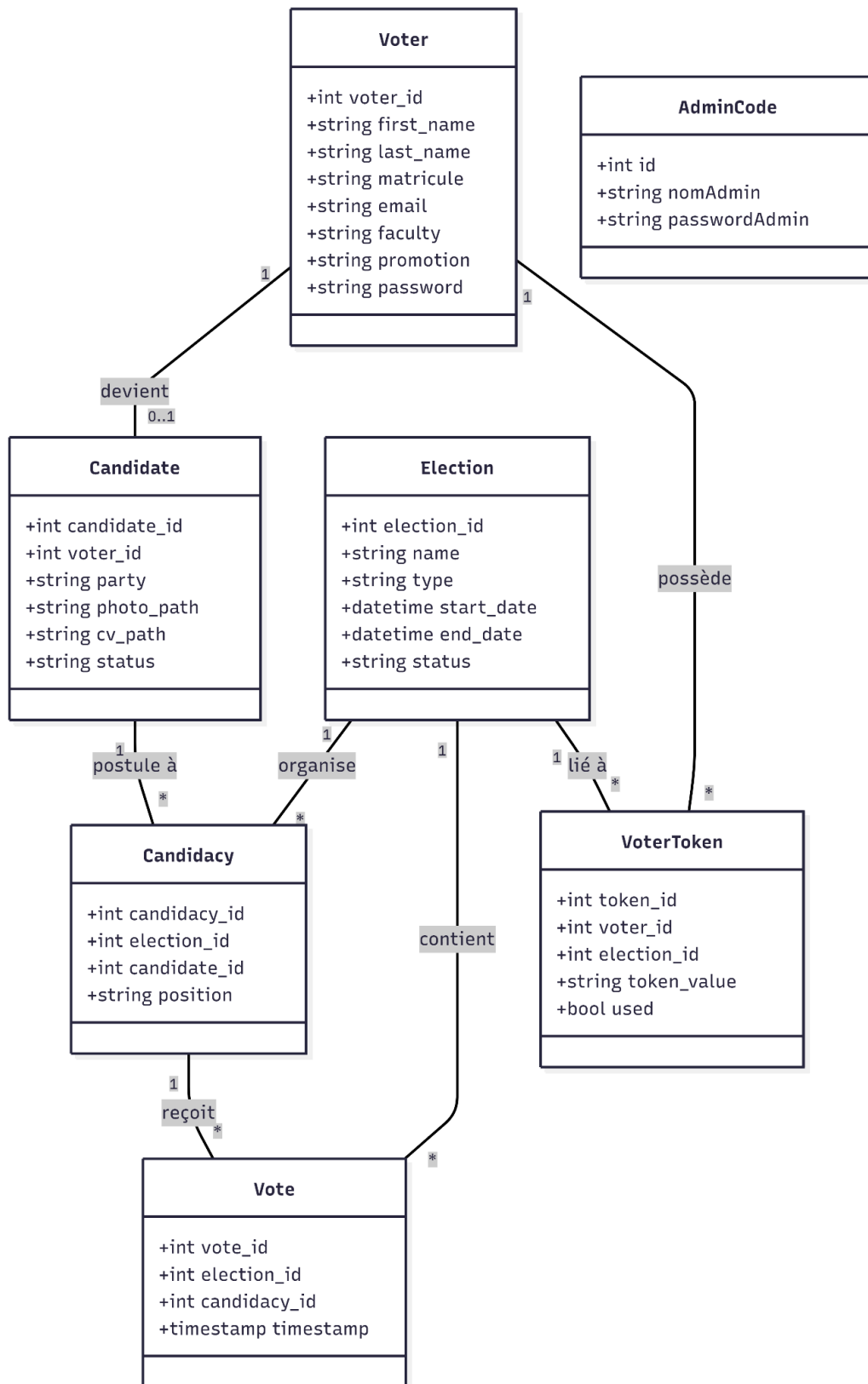


Figure 14 diagramme de classe

2.4.4 Diagramme de déploiement

Le diagramme de déploiement se trouvant à la figure 15 modélise l'architecture physique du système, montrant où les composants logiciels sont exécutés et comment ils communiquent.

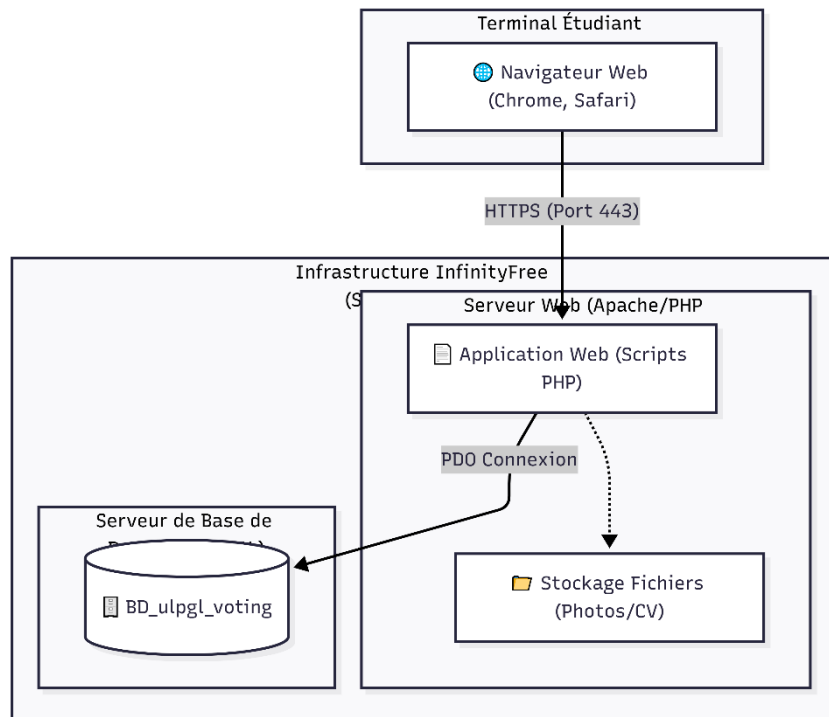


Figure 15 diagramme de déploiement

Conclusion partielle

Ce deuxième chapitre a été consacré à l'approche conceptuelle du système de vote électronique proposé dans le cadre de ce travail. Il avait pour objectif principal de traduire les besoins identifiés et les fondements théoriques présentés au chapitre précédent en une solution technique structurée, cohérente et exploitable.

À travers la présentation des technologies retenues, il a été démontré que le choix d'une architecture web client–serveur basée sur des technologies telles que HTML, CSS, JavaScript, PHP et MySQL est pertinent pour le développement d'une application de vote en ligne fiable, rapide et sécurisée. Ces technologies, largement éprouvées et accessibles, offrent un bon compromis entre performance, flexibilité et facilité de maintenance, tout en étant adaptées au contexte universitaire de l'ULPGL.

Les tableaux de spécifications fonctionnelles et non fonctionnelles ont permis de définir avec précision les exigences du système. Les spécifications fonctionnelles ont clarifié les différentes fonctionnalités attendues, allant de l'authentification des utilisateurs à la gestion des candidatures, des scrutins, du vote et du calcul des résultats. Quant aux spécifications non fonctionnelles, elles ont mis en évidence les contraintes essentielles liées à la sécurité, à la performance, à la disponibilité, à l'ergonomie et à la protection des données, éléments indispensables pour garantir la crédibilité et l'acceptabilité du système.

Par ailleurs, l'utilisation du langage UML pour la modélisation du système a permis de représenter de manière claire et structurée les interactions entre les acteurs, les processus clés et la structure interne de l'application. Les diagrammes de cas d'utilisation, de séquence, de classes et de déploiement constituent ainsi une base solide pour guider la phase de développement et réduire les ambiguïtés lors de l'implémentation.

En somme, ce chapitre a établi les fondations techniques et conceptuelles nécessaires à la réalisation du système de vote électronique. Il a servi de lien essentiel entre l'analyse théorique du

problème et la mise en œuvre pratique de la solution. Sur la base de ces éléments, le chapitre suivant sera consacré à l'implémentation proprement dite du système, en présentant la réalisation des différentes fonctionnalités, l'interface utilisateur, les mécanismes de sécurité ainsi que les tests de validation.

Chapitre 3

IMPLEMENTATION

Ce chapitre est consacré à l'implémentation pratique du système de vote électronique conçu dans le cadre de ce travail. Après avoir établi les bases théoriques du vote électronique au chapitre 1 et défini l'architecture conceptuelle ainsi que les spécifications du système au chapitre 2, il est maintenant question de décrire en profondeur la manière dont ces éléments ont été concrètement traduits en une application web fonctionnelle.

L'objectif principal de ce chapitre est de montrer comment les choix technologiques, les diagrammes de modélisation et les exigences fonctionnelles et non fonctionnelles ont guidé la réalisation du système. Il met également en évidence les mécanismes mis en place pour garantir la sécurité, la fiabilité, la rapidité et la transparence du processus de vote. Enfin, ce chapitre présente les tests effectués afin de valider le bon fonctionnement global de l'application.

3.1 Environnement de développement et cadre de réalisation

La réalisation d'une application web de vote électronique nécessite un environnement de développement adapté, capable de simuler les conditions réelles d'exploitation du système. Pour ce faire, nous avons mis en place un environnement local permettant de développer, tester et valider progressivement chaque fonctionnalité avant un éventuel déploiement en production.

Sur le plan matériel, le développement a été effectué sur mon ordinateur personnel disposant de ressources suffisantes pour exécuter simultanément un serveur web, une base de données et un navigateur. Cet environnement correspond aux conditions généralement rencontrées dans les institutions académiques, ce qui garantit la faisabilité du déploiement futur du système au sein de l'ULPGL.

Sur le plan logiciel, l'utilisation de l'outil XAMPP a permis d'installer un serveur Apache, le langage PHP et le système de gestion de base de données MySQL dans un même environnement. Cette configuration facilite le développement et les tests, tout en reproduisant une architecture

client–serveur proche de la réalité. L’éditeur Visual Studio Code a été utilisé pour la rédaction du code source, tandis que les navigateurs web ont servi à tester l’interface utilisateur et le comportement du système.

À la fin de la réalisation de notre système, nous l’avons hébergé sur l’hébergeur gratuit nommé *infinity free* et dont l’adresse est « <https://i-voting-ulpgl.wuaze.com> ».

3.2 Architecture technique et organisation de l’application

L’application développée repose sur une architecture client–serveur, un modèle largement adopté dans le développement des applications web modernes. Cette architecture permet de séparer clairement les responsabilités entre le client, qui gère l’affichage et les interactions avec l’utilisateur, et le serveur, qui traite la logique métier et l’accès aux données.

Du côté client, l’utilisateur accède au système à travers un navigateur web. Les pages HTML constituent la structure des interfaces, tandis que le CSS assure la mise en forme et l’harmonisation graphique. Le JavaScript intervient pour rendre l’interface dynamique, notamment lors de la validation des formulaires, de l’affichage des messages de confirmation et de certaines interactions sans rechargement de page.

Du côté serveur, les scripts PHP jouent un rôle central. Ils reçoivent les requêtes envoyées par le client, vérifient les droits d’accès, appliquent les règles métier et interagissent avec la base de données MySQL. Cette séparation permet de renforcer la sécurité du système, car les traitements sensibles (authentification, vote, calcul des résultats) sont entièrement réalisés côté serveur, hors de portée directe de l’utilisateur.

L’organisation interne du projet a été pensée de manière modulaire afin de faciliter la maintenance et l’évolution future du système. Les fichiers sont regroupés par fonctionnalité, ce qui permet d’isoler les pages publiques, les pages administrateur, les fichiers de configuration et les ressources statiques. Cette structuration améliore la lisibilité du code et réduit les risques d’erreurs lors des mises à jour.

3.3 Implémentation du module d'inscription

L'inscription constitue l'un des éléments les plus critiques du système, car elle conditionne l'accès aux différentes fonctionnalités et garantit que seuls les utilisateurs autorisés peuvent voter.

Le processus d'inscription illustré à la figure 16 a été conçu pour répondre à deux exigences majeures : la fiabilité de l'identification et la simplicité d'utilisation. Lors de la première étape, l'étudiant est invité à fournir son nom, son matricule et son adresse e-mail. Ces informations sont comparées à celles enregistrées dans la base de données issue de l'université. Cette vérification permet de s'assurer que l'utilisateur est bien un étudiant inscrit et éligible au vote.

Une fois cette vérification effectuée, le système génère un mot de passe temporaire qui est envoyé à l'adresse e-mail fournie. Cette seconde étape renforce la sécurité du système en ajoutant une forme d'authentification à deux facteurs. L'étudiant doit ensuite se connecter en utilisant ce mot de passe pour accéder à son espace personnel. Ce mécanisme permet de limiter les risques d'usurpation d'identité et d'accès frauduleux au système.

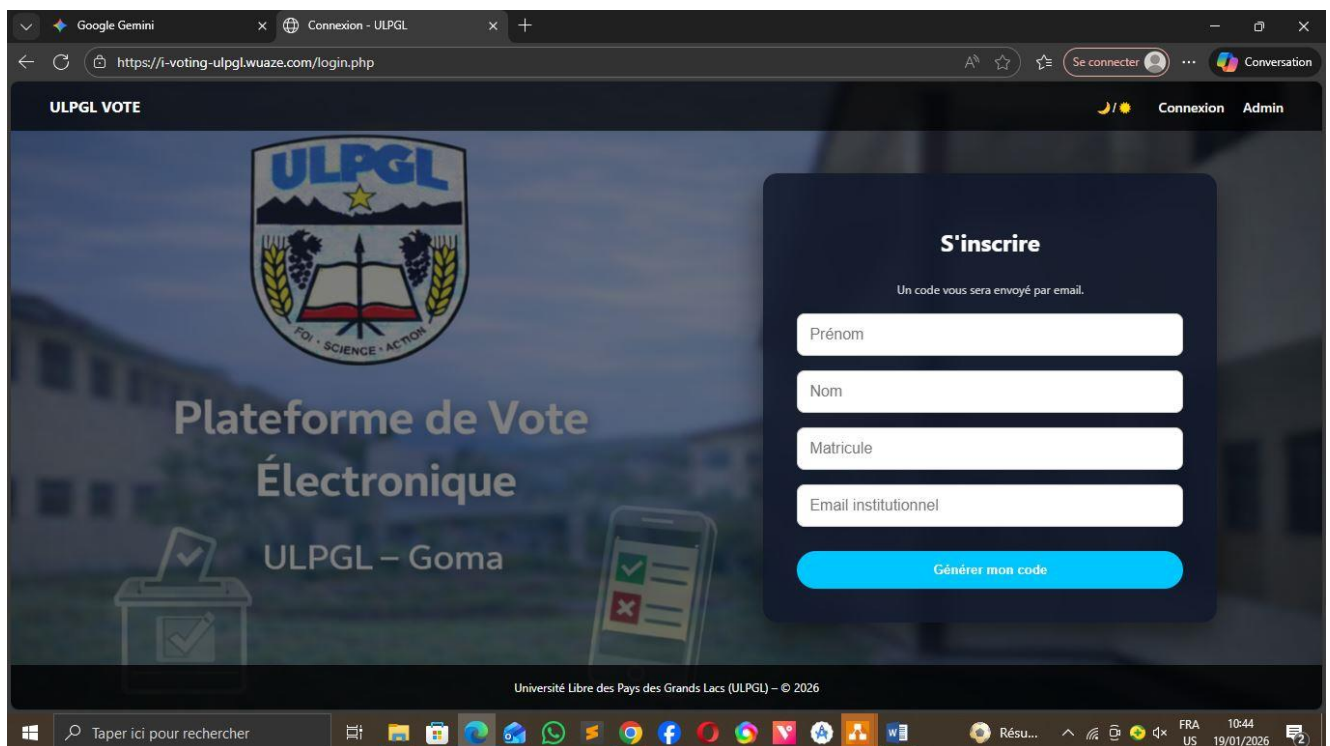
The image is a screenshot of a web browser displaying the login page of the 'ULPGL VOTE' platform. The browser's address bar shows the URL 'https://i-voting-ulpgl.wuaze.com/login.php'. The page features a dark blue header with the text 'ULPGL VOTE' on the left and 'Connexion Admin' on the right. The main content area has a background image of a building and a ballot box. On the left, there is a large logo for 'ULPGL' (Université Libre des Pays des Grands Lacs) with the tagline 'POI - SCIENCE - ACTION' and the text 'Plateforme de Vote Électronique ULPGL - Goma'. On the right, there is a dark blue box titled 'S'inscrire' (Sign up) with the subtext 'Un code vous sera envoyé par email.' Below this, there are four white input fields for 'Prénom', 'Nom', 'Matricule', and 'Email institutionnel'. At the bottom of this box is a blue button labeled 'Générer mon code'. The footer of the page reads 'Université Libre des Pays des Grands Lacs (ULPGL) - © 2026'. The Windows taskbar is visible at the bottom of the screen.

Figure 16 page d'inscription et de connexion

3.4 Implémentation du module de gestion des candidatures

Le module de gestion des candidatures illustré à la figure 18 permet aux étudiants souhaitant se présenter à une élection de soumettre leur dossier de candidature de manière électronique. Cette fonctionnalité vise à remplacer les procédures manuelles souvent longues et peu structurées.

Lors de la soumission d'une candidature, le système demande au candidat de fournir plusieurs informations essentielles, notamment son identité, une photo, un curriculum vitae et une lettre de motivation. Ces éléments sont stockés de manière sécurisée sur le serveur, tandis que les informations descriptives sont enregistrées dans la base de données.

Après la soumission, la candidature est placée dans un état « en attente de validation ». L'administrateur du système peut alors consulter l'ensemble des candidatures reçues, analyser les dossiers et décider de les valider ou de les rejeter comme cela est bien démontré sur la figure 17. Cette étape permet de garantir que seuls les candidats respectant les critères définis participent au scrutin, renforçant ainsi la crédibilité du processus électoral.

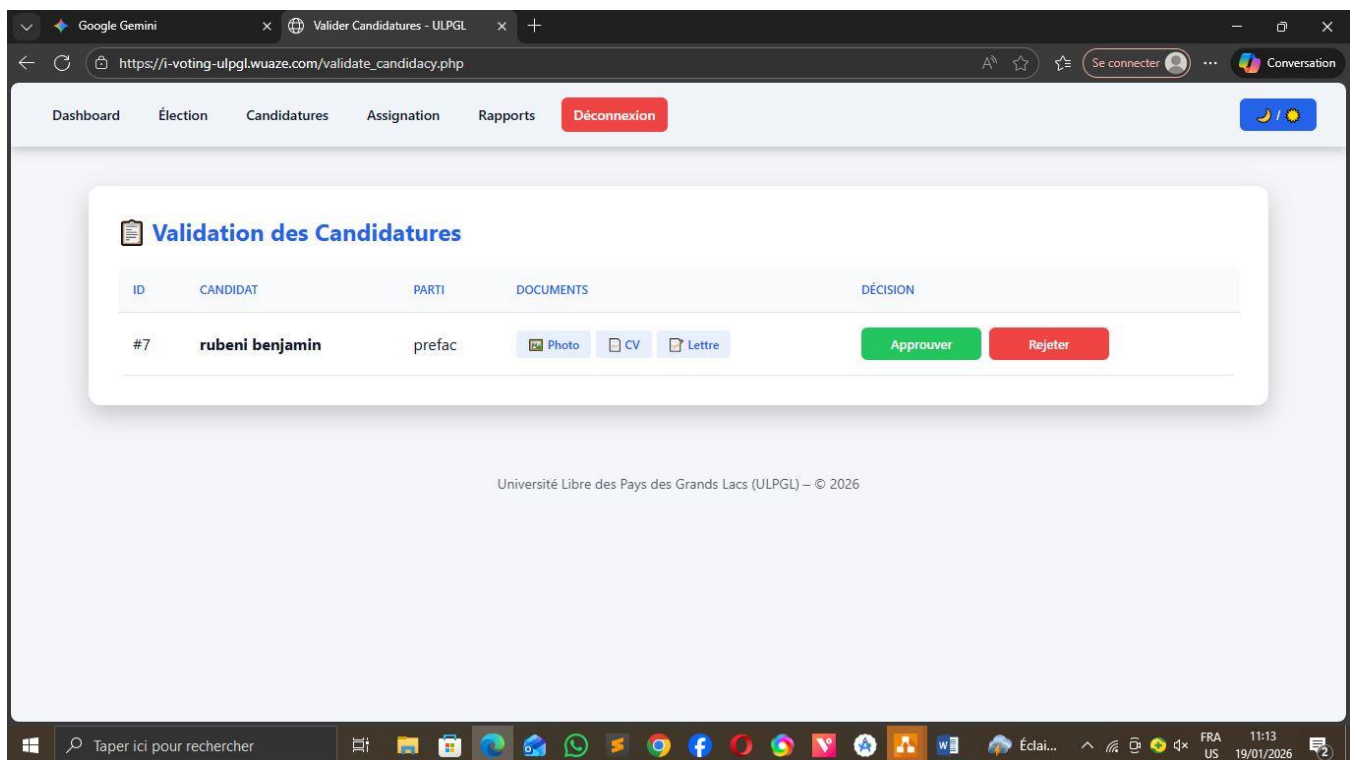


Figure 17 page validation de candidature

Google Gemini x Soumettre Candidature - ULPGL x +

← ↻ 🔒 https://i-voting-ulppl.wuaze.com/submit_candidacy.php A ☆ ☆ Se connecter Conversation

Dashboard Candidature Résultats Déconnexion

Soumettre Candidature

Prénom: Nom:

Parti (Optionnel): Faculté:

Promotion:

Biographie / Programme:

Photo (Profil): Aucun fichier n'a été sélectionné

CV (PDF): Aucun fichier n'a été sélectionné

Lettre de Motivation:

Taper ici pour rechercher

22°C FRA US 11:43 19/01/2026

Figure 18 page soumission de candidature

3.5 Implémentation du module de gestion des scrutins

La gestion des élections illustré à la figure 19 est entièrement assurée par l'administrateur du système. Ce module permet de créer, paramétrer et contrôler l'ensemble du processus électoral. Lors de la création d'une élection, l'administrateur renseigne plusieurs informations importantes, telles que le titre de l'élection, sa description, le type d'élection concerné, ainsi que les dates de début et de fin du vote. Ces paramètres permettent au système de déterminer automatiquement les périodes pendant lesquelles le vote est autorisé.

L'administrateur associe ensuite les candidats validés à l'élection concerné et définit la liste des électeurs éligibles comme cela est décrit sur la figure 20. Cette fonctionnalité est essentielle pour garantir que seuls les étudiants concernés par une élection donnée puissent y participer. Elle permet également d'organiser plusieurs scrutins différents au sein de la même plateforme.

Google Gemini x Créer Élection - ULPGL x +

https://i-voting-ulpgl.wuaze.com/create_election.php

Se connecter Conversation

Dashboard Élection Candidatures Assignment Rapports Déconnexion

Créer une nouvelle Élection

Titre de l'Élection
ex: Élection Porte-Parole 2024

Type de Vote
Sélectionnez un type

Description (Objectifs, règles...)

Date de Début
jj/mm/aaaa --:--

Date de Fin
jj/mm/aaaa --:--

Créer l'Élection

Liste des Élections

Taper ici pour rechercher

20°C FRA US 10:50 19/01/2026

Figure 19 page création d'une élection

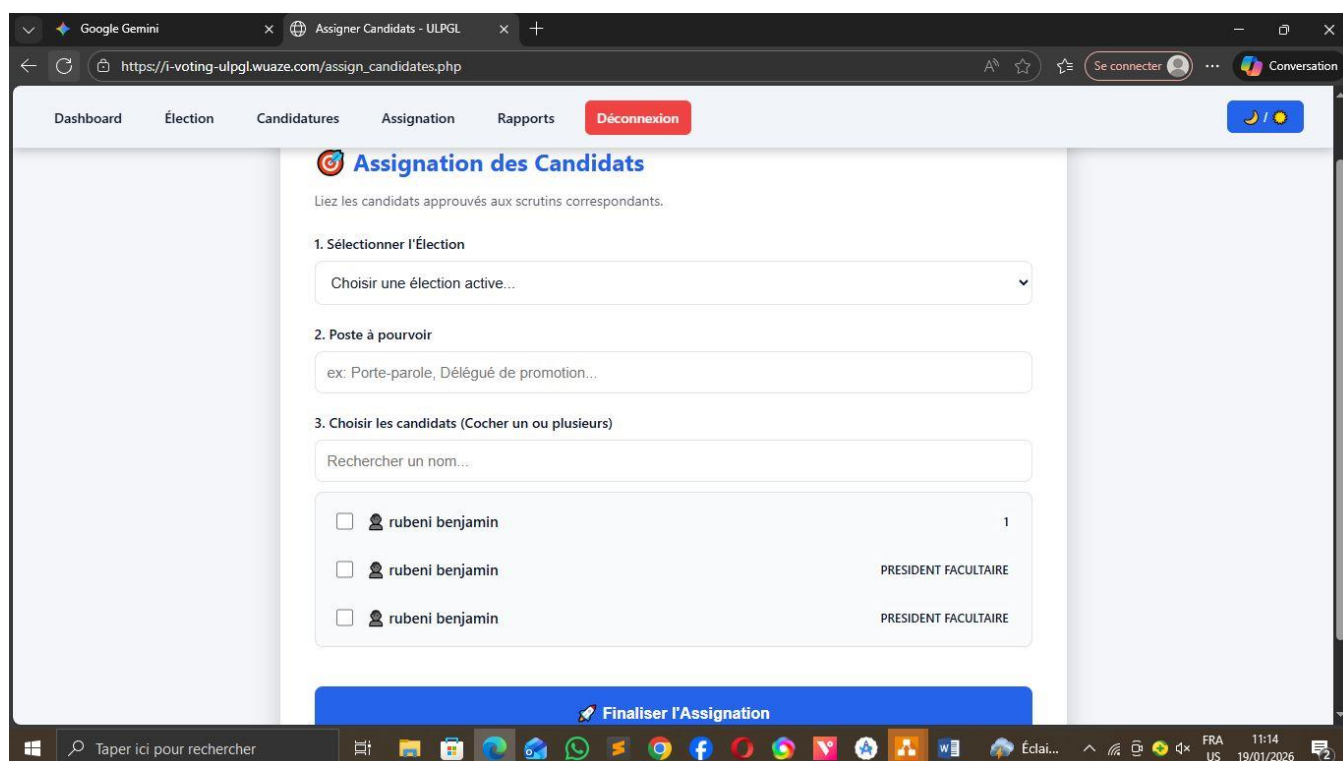


Figure 20 page assignation des candidats a une élection

3.6 Implémentation du module de vote électronique

Le module de vote représenté à la figure 21 est le cœur du système. Il a été conçu pour être simple à utiliser tout en respectant les principes fondamentaux du vote démocratique, notamment le secret et l'unicité du vote.

Une fois authentifié, l'électeur accède à la liste des scrutins en cours pour lesquels il est éligible. Il sélectionne le scrutin souhaité, consulte la liste des candidats et effectue son choix. Avant la soumission définitive, le système affiche une étape de confirmation afin d'éviter les erreurs de sélection.

Lorsque le vote est soumis, le système enregistre le bulletin de manière anonyme dans la base de données. Aucune information permettant de relier directement l'identité de l'électeur à son choix n'est conservée. Le statut de l'électeur est ensuite mis à jour afin d'empêcher toute tentative de

double vote. Un message de confirmation est enfin affiché pour informer l'utilisateur que son vote a été pris en compte avec succès.

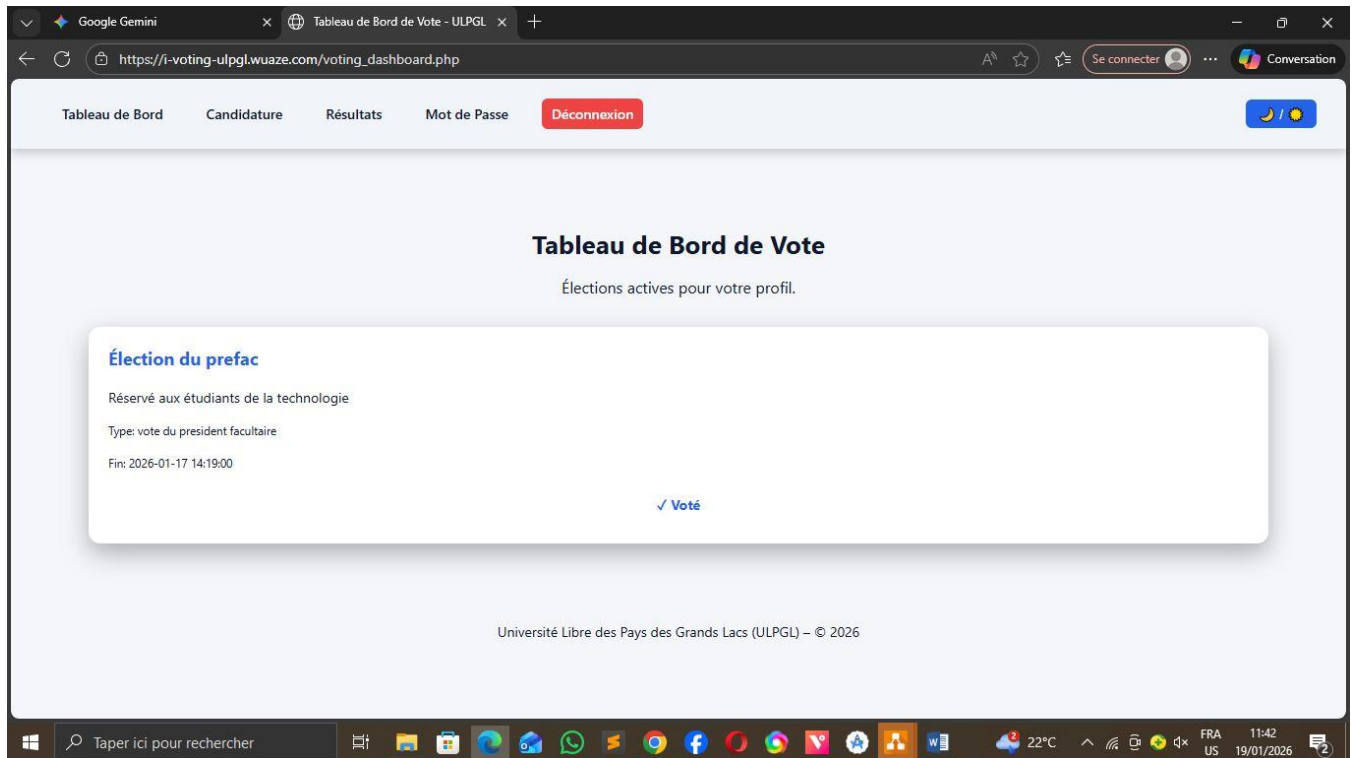


Figure 21 page de vote

3.7 Calcul automatique et affichage des résultats

Au cours de l'élection, le système procède automatiquement au dépouillement des votes. Cette étape repose sur des requêtes SQL permettant de compter le nombre de voix obtenues par chaque candidat comme cela est illustré sur la figure 22.

Les résultats sont ensuite présentés sous forme de tableaux récapitulatifs, facilitant ainsi leur lecture et leur interprétation. Ce mécanisme permet d'éliminer les erreurs humaines liées au dépouillement manuel et d'accélérer considérablement la proclamation des résultats.

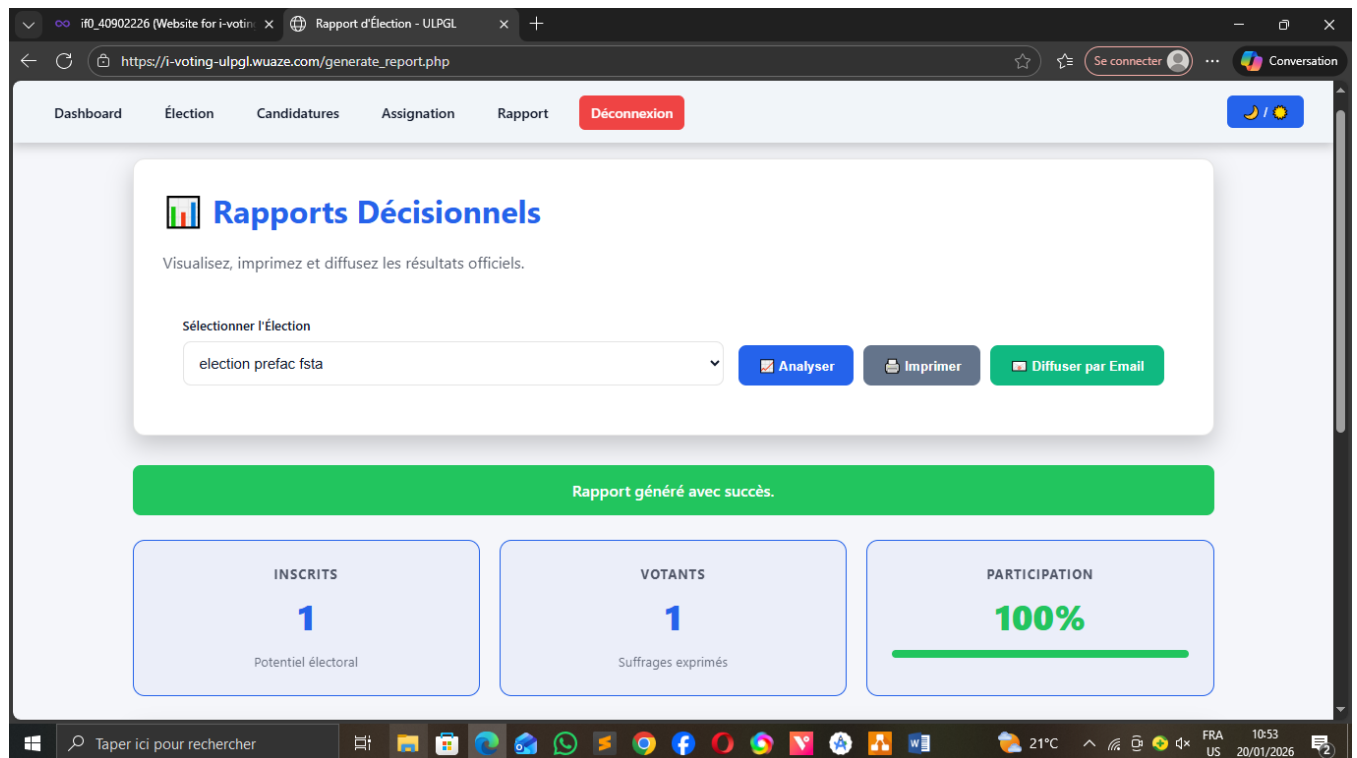


Figure 22 page affichage des résultats du vote

3.8 Sécurité du système et protection des données

La sécurité a été prise en compte à toutes les étapes de l'implémentation. Les entrées utilisateur sont systématiquement contrôlées afin d'éviter les attaques par injection SQL ou par scripts malveillants. Les requêtes préparées sont utilisées pour renforcer la protection de la base de données.

Les votes sont stockés de manière anonymisé et séparée des informations personnelles des électeurs. De plus, l'accès aux différentes fonctionnalités du système est strictement contrôlé en fonction du rôle de l'utilisateur (électeur ou administrateur).

3.9 Tests, validation et fiabilité du système

Afin de garantir la fiabilité du système, plusieurs tests ont été réalisés. Les tests fonctionnels ont permis de vérifier que chaque module répond correctement aux exigences définies dans les

spécifications mais des tests de charge n'ont pas été effectués pour observer le comportement du système en cas de connexions multiples et simultanées.

Les résultats obtenus montrent que le système fonctionne de manière stable et fiable, et qu'il est capable de gérer un processus de vote électronique dans des conditions réelles d'utilisation.

Conclusion partielle

Ce chapitre a présenté de manière détaillée l'implémentation du système de vote électronique développé dans le cadre de ce travail. Il a permis de démontrer que la solution proposée a été techniquement réalisée et adaptée au contexte universitaire de l'ULPGL. Les fonctionnalités implémentées répondent aux besoins identifiés et constituent une alternative moderne et fiable au système de vote traditionnel.

Conclusion générale

Au terme de ce travail intitulé « *Conception et réalisation d'une application web fiable, rapide et sécurisée pour la gestion de vote de porte-parole des étudiants, présidents facultaires et chefs de promotion au sein de l'ULPGL Goma* », il convient de dresser un bilan global des résultats obtenus et d'évaluer l'atteinte des objectifs fixés.

Dans un contexte marqué par la digitalisation croissante des services et par la nécessité de renforcer la transparence et l'efficacité des processus décisionnels, ce travail s'est inscrit dans une dynamique de modernisation du système électoral étudiant. En effet, le système de vote traditionnel utilisé à l'ULPGL Goma présentait plusieurs insuffisances, notamment la lenteur du processus, la vulnérabilité aux erreurs humaines et aux fraudes, ainsi qu'un faible taux de participation des étudiants. Ces constats ont justifié la nécessité de proposer une solution informatique adaptée, fiable et sécurisée.

L'objectif général de ce travail était de concevoir et de réaliser un système informatique permettant la gestion efficace des votes des représentants des étudiants au sein de l'ULPGL. Pour atteindre cet objectif, plusieurs objectifs spécifiques ont été définis, allant de l'analyse des besoins fonctionnels et non fonctionnels à l'implémentation d'une application web intégrant des mécanismes de sécurité, d'authentification et de dépouillement automatique des votes. L'ensemble de ces objectifs a été atteint de manière satisfaisante.

En ce qui concerne les réponses à nos questions de recherche, ce travail a permis de confirmer que l'intégration d'une plateforme web centralisée résout efficacement les lacunes du système de vote manuel à l'ULPGL. A la question de savoir comment garantir l'intégrité du scrutin, nous avons apporté une réponse par la mise en place de mécanismes d'authentification unique et de protocole de sécurité interdisant le double vote. Quant à l'accessibilité et à la rapidité du traitement des données, l'automatisation du dépouillement et la disponibilité de l'interface sur divers supports numériques offrent une réponse pragmatique, réduisant drastiquement le temps d'attente et les risques d'erreurs humaines. Ainsi le système conçu ne se contente pas de numériser le vote, mais il sécurise chaque étape du processus électoral, validant ainsi nos hypothèses de départ.

Sur le plan méthodologique, l'adoption d'une approche centrée sur l'utilisateur combinée à la méthodologie agile a permis de mieux cerner les attentes des différentes parties prenantes et de

développer une solution évolutive et adaptée au contexte universitaire. La modélisation du système à l'aide du langage UML a facilité la compréhension globale de l'architecture et des interactions entre les différents acteurs du système.

Sur le plan technique, l'utilisation des technologies web telles que HTML, CSS, JavaScript, PHP et MySQL a permis de concevoir une application web fonctionnelle, intuitive et accessible depuis différents supports. Les mécanismes mis en place pour l'authentification des utilisateurs, la gestion des rôles, la prévention du double vote et la sécurisation des données contribuent à renforcer l'intégrité, la fiabilité et la crédibilité du processus électoral. De plus, le dépouillement automatique et la proclamation quasi instantanée des résultats constituent une avancée majeure par rapport au système manuel précédemment utilisé.

Les résultats obtenus démontrent que la solution proposée répond efficacement aux problématiques identifiées au départ et confirme les hypothèses formulées. L'application développée permet non seulement d'améliorer la gestion des élections étudiantes, mais aussi de favoriser la participation, la transparence et la confiance des étudiants dans le processus électoral.

Difficultés rencontrées

La réalisation de ce travail n'a pas été exempte de difficultés. Parmi les principales contraintes rencontrées figurent les limites temporelles liées au calendrier académique, qui ont restreint la durée consacrée à la réalisation du système, aux tests approfondis du système mais aussi notons que jusque-là le système conçu n'a pas encore été testé dans des conditions réelles. À cela s'ajoutent les contraintes techniques, notamment la gestion de la sécurité des données, la prévention du double vote et l'implémentation d'un mécanisme d'authentification fiable, qui ont nécessité des choix techniques rigoureux et parfois complexes. Par ailleurs, les difficultés liées à l'instabilité de la connexion Internet et aux coupures d'électricité, fréquentes dans le contexte local, ont constitué un défi supplémentaire lors des phases de développement et de test. Précisons aussi que tout au long de l'élaboration de notre travail, nous étions censés, faire de cours, passer des examens, penser aussi à la réalisation de notre système, cela a été parmi les grandes difficultés parce que combiner toutes ces tâches la n'étaient pas vraiment facile.

Améliorations possibles et perspectives

Bien que l'application développée réponde aux objectifs fixés, plusieurs améliorations peuvent être envisagées pour renforcer davantage ses performances et sa sécurité. Il serait notamment pertinent d'intégrer des mécanismes d'authentification avancés, tels que l'authentification biométrique ou l'envoi de codes à usage unique (OTP) via SMS. De plus, la mise en place d'un système d'audit et de journalisation des actions permettrait d'améliorer la traçabilité et la transparence du processus électoral. À long terme, l'intégration de technologies émergentes, telles que la blockchain, pourrait également être explorée afin de garantir une immutabilité accrue des votes et de renforcer la confiance des utilisateurs.

Recommandations

Au regard des résultats obtenus, il est recommandé aux autorités universitaires de l'ULPGL d'envisager l'adoption progressive de ce système de vote électronique afin d'évaluer son efficacité en conditions réelles. Il est également recommandé de sensibiliser et de former les étudiants et le personnel administratif à l'utilisation de la plateforme afin de favoriser son appropriation et d'optimiser son impact. Enfin, il est conseillé d'assurer une maintenance régulière du système et de procéder à des mises à jour continues pour faire face aux nouvelles menaces de sécurité et aux évolutions technologiques.

En définitive, ce travail constitue juste une contribution à l'amélioration du processus électoral en milieu universitaire. Il démontre que l'usage raisonné des technologies de l'information et de la communication peut apporter des solutions concrètes aux défis organisationnels et démocratiques. Ainsi, les objectifs poursuivis dans le cadre de ce travail ont été pleinement atteints, et la solution proposée peut servir de base pour de futures améliorations et recherches dans le domaine du vote électronique.

Bibliographie

- [1] M. B. Victor, «vote électronique qu'en sein de l'ULPGL,» ULPGL, GOMA, 2020.
- [2] M. C. Frederick, «technologie blockchain pour le vote en RDC,» ULPGL, GOMA, 2023.
- [3] A. Seye, *Vote électronique*, Alger, 2019.
- [4] M. J. Radwin, «An untraceable, universally verifiable voting scheme,» *voting*, 1995.
- [5] j. k. e. r. g. kazuo sako, «secure electronic elections,» 1999.
- [6] D. Chaum, «Blind signatures for untraceable payments,» acte de la conférence de crypto , 1982.
- [7] C. University, «Une analyse comparative du vote électronique,» 2010.
- [8] Y. Izzeddine, «Conception et réalisation d'un système de vote à distance,» UNIVERSITE MOHAMED BOUDIAF M'SILA, Alger, 2019.
- [9] B. Adida, «Helios : web-based open audit voting,» *usenix security symposium*, 2008.
- [10] s. g. e. v. l. veronique cortier, «Belenios,» *USENIX/ACM workshop on cyber security experimentation and test*, 2014.
- [11] w. j. g. e. a. Ralf phillipp, «security analysis of the estonian internet voting system,» 2014.
- [12] c. c. e. a. j. alex halderman, «the danger of sending votes over the internet: attacking the NSW ivote system,» 2015.
- [13] v. e. a. rutsche, «analysis of the norwegian e-voting protocol,» 2014.
- [14] m. nebra, *apprendre à créer votre site web avec html5 et css3*, 2012.
- [15] m. nebra, *apprendre à créer votre site web avec html5 et css3*, 2012.
- [16] m. haverbeke, *eloquent javascript*, 2024.
- [17] s. d. zeros, *apprendre PHP pour les zeros*, 2004.
- [18] *des bases de données en python avec sqlite3*, 2019.
- [19] p. roques, *UML par la pratique*, 2010.

[20] JUnit.org. [En ligne]. Available: <http://www.junit.org>. [Accès le 2 Janvier 2019].

[21]

Annexes

1. Code source de la page login

```
<?php
session_start();
require_once 'config.php';
require_once 'includes/functions.php';

// --- AJOUT DES CLASSES PHPMailer ---
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'includes/PHPMailer/Exception.php';
require 'includes/PHPMailer/PHPMailer.php';
require 'includes/PHPMailer/SMTP.php';

$message = "";
$message_type = 'error';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $login_type = $_POST['login_type'];
```

```

// --- ADMIN ---
if ($login_type == 'admin') {
    $username = htmlspecialchars($_POST['username']);
    $password = $_POST['password'];
    $stmt = $pdo->prepare("SELECT * FROM voter WHERE nomAdmin = ? AND
passwordAdmin = ?");
    $stmt->execute([$username, $password]);
    $admin = $stmt->fetch();
    if ($admin) {
        $_SESSION['admin'] = true;
        header('Location: admin_dashboard.php');
        exit;
    } else { $message = 'Identifiants invalides.'; }
}

// --- ÉTUDIANT ---
elseif ($login_type == 'student') {
    if (isset($_POST['action']) && $_POST['action'] == 'inscrire') {
        $first_name = htmlspecialchars($_POST['first_name']);
        $last_name = htmlspecialchars($_POST['last_name']);
        $matricule = htmlspecialchars($_POST['matricule']);
        $email = htmlspecialchars($_POST['email']);

        $stmt = $pdo->prepare("SELECT voter_id FROM voter WHERE first_name = ? AND
last_name = ? AND matricule = ? AND email = ?");
        $stmt->execute([$first_name, $last_name, $matricule, $email]);
        $voter = $stmt->fetch();

        if ($voter) {

```

```

$generated_code = substr(str_shuffle(str_repeat("0123456789ABCDEF", 4)), 0, 4);
$update = $pdo->prepare("UPDATE voter SET login_code = ? WHERE voter_id = ?");

if ($update->execute([$generated_code, $voter['voter_id']])) {

    // --- CONFIGURATION D'EN      CAL VIA SMTP ---
    $mail = new PHPMailer(true);
    try {
        $mail->isSMTP();
        $mail->Host      = 'smtp.gmail.com';
        $mail->SMTPAuth  = true;
        $mail->Username  = 'rubenibenjamin4@gmail.com';
        $mail->Password  = 'notmrkiguqmfuiwq';
        $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
        $mail->Port      = 587;

        $mail->setFrom('no-reply@ulppl.net', 'ULPGL VOTE');
        $mail->addAddress($email);

        $mail->isHTML(true);
        $mail->Subject = "Votre code de vote ULPGL";
        $mail->Body    = "Bonjour <b>$first_name</b>,<br><br>Votre code de connexion
est : <b style='font-size:18px; color:#00c6ff;'>$generated_code</b><br><br>Utilisez votre matricule
pour vous connecter.";

        $mail->send();
        $message = "Code envoyé avec succès à votre adresse email.";
        $message_type = "success";
    } catch (Exception $e) {
        // En cas d'erreur SMTP, on affiche le code pour permettre le test localement

```

```

        $message = "Serveur local : Votre code est $generated_code (Erreur Mail: {$mail-
>ErrorInfo})";
        $message_type = "success";
    }
}
} else { $message = "Données incorrectes : Votateur non répertorié."; }
}
elseif (isset($_POST['action']) && $_POST['action'] == 'login_code_modal') {
    $matricule = htmlspecialchars($_POST['matricule_modal']);
    $code = htmlspecialchars($_POST['code_modal']);
    $stmt = $pdo->prepare("SELECT voter_id, login_code FROM voter WHERE matricule =
?");
    $stmt->execute([$matricule]);
    $voter = $stmt->fetch();

    if ($voter && $voter['login_code'] === $code) {
        $_SESSION['voter_id'] = $voter['voter_id'];
        header('Location: voting_dashboard.php');
        exit;
    } else { $message = 'Matricule ou code incorrect.'; }
}
}
}
?>

```

```

<!DOCTYPE html>

```

```

<html lang="fr">

```

```

<head>

```

```

    <meta charset="UTF-8">

```

```

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<title>Connexion - ULPGL</title>
<style>
  :root{
    --primary: #00c6ff;
    --card-dark: rgba(15, 23, 42, 0.85);
    --card-light: rgba(255, 255, 255, 0.9)
  }

  body {
    min-height: 100vh;
    margin: 0;
    font-family: 'Segoe UI', sans-serif;
    background: linear-gradient(rgba(0,0,0,0.5), rgba(0,0,0,0.5)), url('../public/css/ul.png') no-
repeat center center fixed;
    background-size: cover;
    display: flex;
    flex-direction: column;
    transition: 0.3s;
  }

  body.light { background: linear-gradient(rgba(255,255,255,0.6), rgba(255,255,255,0.6)),
url('../public/css/ul.png') no-repeat center center fixed; background-size: cover; }

  nav {
    padding: 15px 40px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    background: rgba(0,0,0,0.7);
    color: white;

```

```

}
body.light nav { background: white; color: black; border-bottom: 1px solid #ddd; }

.nav-links a { color: inherit; text-decoration: none; margin-left: 15px; font-weight: 600; font-size: 14px; }

.login-container {
  flex: 1;
  display: flex;
  justify-content: flex-end;
  align-items: center;
  padding: 0 8% 0 0;
}

.login-form {
  width: 100%;
  max-width: 400px;
  background: var(--card-dark);
  color: white;
  padding: 35px;
  border-radius: 15px;
  backdrop-filter: blur(10px);
  box-shadow: 0 15px 35px rgba(0,0,0,0.5);
}

body.light .login-form { background: var(--card-light); color: #0f172a; box-shadow: 0 15px 35px rgba(0,0,0,0.1); }

h2 { text-align: center; margin-bottom: 25px; }

```

```
.message { padding: 12px; border-radius: 8px; text-align: center; margin-bottom: 20px; font-size: 14px; }
```

```
.message.error { background: #ef4444; color: white; }
```

```
.message.success { background: #10b981; color: white; }
```

```
label { display: block; margin-top: 15px; font-size: 14px; font-weight: 500; }
```

```
input { width: 100%; padding: 12px; margin-top: 5px; border-radius: 8px; border: 1px solid #ccc; box-sizing: border-box; font-size: 16px; }
```

```
.submit-btn { width: 100%; padding: 12px; margin-top: 25px; border: none; border-radius: 25px; background: var(--primary); color: white; font-weight: bold; cursor: pointer; transition: 0.3s; }
```

```
.submit-btn:hover { filter: brightness(1.1); }
```

```
footer { padding: 15px; text-align: center; font-size: 12px; background: rgba(0,0,0,0.8); color: white; }
```

```
@media (max-width: 850px) {
```

```
  .login-container { justify-content: center; padding: 20px; }
```

```
  nav { flex-direction: column; gap: 10px; }
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<nav>
```

```
<div class="logo"><strong>ULPGL VOTE</strong></div>
```

```
<div class="nav-links">
```

```
<button onclick="toggleTheme()" style="cursor:pointer; border:none; background:none; color:inherit;"> 🌙/☀️</button>
```

```

<?php if (!isset($_GET['type'])): ?>
    <a href="?type=student">Connexion</a>
    <a href="?type=admin">Admin</a>
<?php else: ?>
    <a href="login.php">Inscription</a>
<?php endif; ?>
</div>
</nav>

<div class="login-container">
    <div class="login-form">
        <?php if($message): ?>
            <div class="message <?= $message_type ?>"><?= $message ?></div>
        <?php endif; ?>

        <?php if (isset($_GET['type']) && $_GET['type']=='admin'): ?>
            <form method="POST">
                <h2>Admin</h2>
                <input type="hidden" name="login_type" value="admin">
                <label>Nom d'utilisateur</label>
                <input type="text" name="username" required>
                <label>Mot de passe</label>
                <input type="password" name="password" required>
                <button class="submit-btn">Connexion</button>
            </form>

        <?php elseif (isset($_GET['type']) && $_GET['type']=='student'): ?>
            <form method="POST">
                <h2>Connexion Étudiant</h2>
                <input type="hidden" name="login_type" value="student">
                <input type="hidden" name="action" value="login_code_modal">

```

```

        <label>Matricule</label>
        <input type="text" name="matricule_modal" required placeholder="Ex: 22ULP001">
        <label>Code (reçu par mail)</label>
        <input type="text" name="code_modal" maxlength="4" required placeholder="Code à
4 chiffres">
        <button class="submit-btn">Se connecter</button>
    </form>
    <?php else: ?>
        <form method="POST">
            <h2>S'inscrire</h2>
            <p style="font-size: 12px; text-align: center; opacity: 0.8;">Un code vous sera envoyé
par email.</p>
            <input type="hidden" name="login_type" value="student">
            <input type="hidden" name="action" value="inscrire">
            <input type="text" name="first_name" placeholder="Prénom" required style="margin-
bottom:10px;">
            <input type="text" name="last_name" placeholder="Nom" required style="margin-
bottom:10px;">
            <input type="text" name="matricule" placeholder="Matricule" required style="margin-
bottom:10px;">
            <input type="email" name="email" placeholder="Email institutionnel" required>
            <button class="submit-btn">Générer mon code</button>
        </form>
    <?php endif: ?>
</div>

<footer>
    Université Libre des Pays des Grands Lacs (ULPGL) – © <?= date('Y') ?>
</footer>

```

```
<script>
  function toggleTheme(){
    const isLight = document.body.classList.toggle('light');
    localStorage.setItem('theme', isLight ? 'light' : 'dark');
  }
  window.onload = () => {
    if(localStorage.getItem('theme') === 'light') document.body.classList.add('light');
  }
</script>
</body>
</html>
```

2. Présentation des tables de la base des données

The screenshot shows the phpMyAdmin interface for a database named 'if0_40902226_ulpgl_voting'. The 'Structure' tab is selected, displaying a list of tables with their respective actions (Browse, Structure, Search, Insert, Empty, Drop). The tables are: admincode, candidacy, candidate, election, logs, vote, voter, and votertoken. A summary row at the bottom indicates 8 tables and a total of 20 MyISAM tables with latin1_swedish_ci collation.

Table	Action	Rows	Type	Collation
☐ admincode		1	InnoDB	utf8mb4_general_ci
☐ candidacy		4	InnoDB	utf8mb4_general_ci
☐ candidate		6	InnoDB	utf8mb4_general_ci
☐ election		2	InnoDB	utf8mb4_general_ci
☐ logs		0	InnoDB	utf8mb4_general_ci
☐ vote		2	InnoDB	utf8mb4_general_ci
☐ voter		3	InnoDB	utf8mb4_general_ci
☐ votertoken		2	InnoDB	utf8mb4_general_ci
8 tables	Sum	20	MyISAM	latin1_swedish_ci

Below the table list, there is a 'Filters' section with a search box and a 'Check all' checkbox. At the bottom, there is a 'Create table' button and a 'Number of columns' input field set to 4.

3. Présentation de l'hébergeur

The screenshot shows a web browser window with multiple tabs. The active tab is 'dash.infinityfree.com/accounts/if0_40902226'. The browser's address bar shows the URL. Below the browser window, a notification bar states 'Google Chrome isn't your default browser' with a 'Set as default' button.

The main content area of the dashboard is titled 'MANAGE IF0_40902226'. It features a sidebar on the left with the following menu items: Overview (selected), Control Panel, File Manager, Script Installer, Account Statistics, FTP Details, MySQL Databases, Deactivation History, and Account Settings.

The main content area is divided into several sections:

- Top Section:** Contains three buttons: 'Website Builder' (blue), 'Script Installer' (purple), and 'File Manager' (orange).
- Domains Section:** Includes a search bar labeled 'Search domain...'. Below it is a table with the following data:

DOMAIN	ACTIONS
i-voting-ulppl.wuaze.com	→ Manage View Site

Below the table is a '+ Add Domain' button.
- Account Details Section:** Displays the following information:
 - USERNAME:** if0_40902226
 - PASSWORD:** masked with asterisks, with copy and toggle visibility icons.
 - STATUS:** Active (indicated by a green badge).
 - LABEL:** Website for i-voting-ulppl.wuaze.com

4.Présentation du gestionnaire des fichiers de infinity free

