UNIVERSITÉ DU BURUNDI
GRENIER DU SAVOIR DU BURUNDI
Dépôt institutionnel officiel
https://repository.ub.edu.bi

Conception et implémentation d'un système automatisé de gestion et optimisation du processus de recensement : « cas du recensement pré-électoral de la commission électorale nationale indépendante du Burundi « CENI »

Bucumi, Jean Claude

2018-01
UB, FSI

https://repository.ub.edu.bi/handle/123456789/230

Document téléchargé depuis le **Grenier du Savoir du Burundi** — https://repository.ub.edu.bi

UNIVERSITE DU BURUNDI



FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT DES TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION (TIC)
OPTION : GENIE INFORMATIQUE

CONCEPTION ET IMPLEMENTATION D'UN SYSTEME AUTOMATISE DE GESTION ET OPTIMISATION DU PROCESSUS DE RECENSEMENT : « CAS DU RECENSEMENT PRE-ELECTORAL DE LA COMMISSION ELECTORALE NATIONALE INDEPENDANTE DU BURUNDI « CENI » »

Projet de fin d'études présenté et défendu publiquement par Jean Claude BUCUMI en vue de l'obtention d'un diplôme de Master en Sciences de l'Ingénieur

Sous la direction de :

Dr. NDIKUMAGENGE Jérémie	Directeur

Membres du Jury

Dr NAHAYO Fulgence	Président
Dr SAHINGUVU William	Vice-Président
Dr MANIRABONA Audace	Secrétaire
Dr NDIKUMAGENGE Jérémie	Membre
Dr NDAYISABA Longin	Membre

Bujumbura, Janvier 2018

Dédicaces

A Dieu Tout Puissant ;

A mon regretté père ;

A ma chère mère ;

A mes frères et sœurs ;

A la famille NDINZEMENSHI Bernard ;

A tous mes oncles et tantes ;

A tous mes cousins et cousines ;

A tous mes neveux et nièces ;

A tous mes camarades de classe ;

A tous mes amis et connaissances ;

A tous ceux qui me sont chers ;

Jean Claude BUCUMI

Remerciements

Au terme de ce travail, je tiens à remercier le promoteur et directeur de mon mémoire, Dr. Jérémie NDIKUMAGENGE pour avoir accepté de diriger cette thèse. Son expérience, sa rigueur scientifique, sa disponibilité et son talent pédagogique témoignant d'un savoir étendu m'ont permis de mener à bout ce projet. Je tiens aussi à remercier tous ceux qui ont participé à ma formation depuis l'école primaire jusqu'à l'Université du Burundi. Je tiens particulièrement à exprimer mes sentiments de gratitude aux professeurs du département des Technologies de l'Information et de la Communication (TIC) de la Faculté des Sciences de l'Ingénieur (FSI) de l'Université du Burundi pour leur dévouement ainsi que le savoir-faire qu'ils m'ont dispensé et légué, pour m'avoir surtout fourni tous les outils nécessaires à la réussite de mes études universitaires.

Il m'est aussi agréable d'exprimer ma reconnaissance au personnel de la Commission Electorale Nationale Indépendante plus particulièrement à Madame Fikira Mawazo et Madame Consolée BIZIMANA pour leur accueil chaleureux, leur assistance et leur soutient qu'elles m'ont réservés lors de mes descentes effectuées sur terrain au cours de mes travaux d'exploration et étude de l'existant.

J'exprime ma gratitude, mes remerciements à ma famille qui m'a toujours soutenu tout au long de mes études. Leur soutien et leur engagement sans faille m'ont été d'une importance capitale tout au long de mes études. J'en suis vraiment fier et profondément reconnaissant. Mes expressions et sentiments de gratitude s'adressent également à toute personne qui, de près ou de loin m'a aidé et encouragé au cours de mes études et m'a particulièrement soutenu et appuyé tout au long de la période dédiée aux travaux de ce projet de recherche.

Enfin mes remerciements sont aussi adressés à tous les mes membres du jury qui ont accepté de lire et participé à l'évaluation de ce travail.

Liste des sigles et abréviations

AC	Autorité de Certification
ACL	Access Control List
ANSSI	Agence Nationale de la Sécurité des Systèmes d'Information
AOP	Aspect Oriented Programming
API	Application Programming Interface
CECI	Commission Electorale Communale Indépendante
CENI	Commission Electorale Nationale Indépendante
CEPI	Commission Electorale Provinciale Indépendante
CSRF	Cross Site Request Forgery
CSS	Cascading Style Sheet
CTD	Centre de Traitement des Données
DAO	Data Access Object
EJB	Enterprise Java Bean
FSI	Faculté des Sciences de l'Ingénieur
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IMAP	Internet Message Access Protocol
IoC	Inversion of Control
JDBC	Java Database Connectivity
JPA	Java Persistence API
JTA	Java Transaction API
LDAP	Lightweight Directory Access Protocol
MVC	Modèle-Vue-Contrôleur
NAXSI	Nginx Anti XSS & SQL Injection
NGINX	Engine-X
NTIC	Nouvelles Technologies de l'Information et de la Communication
OMG	Object Management Group
OOSE	Object Oriented Software Engineering
OS	Operating System
OWASP	Open Web Application Security Project
PACE	Projet d'Appui au Cycle Electoral
PNUD	Programme des Nations Unies pour le Développement
POA	Programmation Orientée Aspect
POJO	Plain Old Java Object
POP3	Post Office Protocol
SGBD	Système de Gestion des Bases de Données

SIGELE	Système Intégré de Gestion des Elections
SQL	Structured Query Language
SSL	Secure Socket Layer
TIC	Technologies de l'Information et de la Communication
UB	Université du Burundi
UML	Unified Modeling Language
W3C	World Wide Web Consortium
WAF	Web Application Firewall
WASC	Web Application Security Consortium
XSS	Cross Site Scripting

Liste des figures et schémas

Figure 1: Organigramme de la CENI 4
Figure 2: Cycle Electoral 6
Figure 3: Présentation du modèle 4 + 1 vues 16
Figure 4: Le cycle de vie du logiciel, le cycle en V 20
Figure 5: Diagramme des cas d'utilisation du nouveau système 23
Figure 6 : Diagramme de séquence pour le cas d'utilisation « Authentification » 34
Figure 7: Diagramme de séquence pour le cas d'utilisation « Inscrire un candidat électeur » 35
Figure 8: Diagramme de séquence pour le cas d'utilisation « Mettre à jour un candidat électeur » 36
Figure 9: Diagramme de séquence pour le cas d'utilisation « Vérifier l'inscription d'un candidat électeur » 37
Figure 10: Diagramme d'activités pour le cas d'utilisation « Authentification » 38
Figure 11: Diagramme d'activités pour le cas d'utilisation « Inscrire un candidat électeur » 39
Figure 12: Diagramme d'activités pour le cas d'utilisation « Mettre à jour un candidat électeur » 40
Figure 13: Diagramme d'activités pour le cas d'utilisation « Vérifier l'inscription d'un candidat électeur » 41
Figure 14: Diagramme des classes 42
Figure 15: Diagramme des composants 43
Figure 16: Diagramme de déploiement 44
Figure 17: Schéma illustratif du modèle du processus de naissance et de mort 48
Figure 18: Schéma illustratif du modèle de files d'attente 50
Figure 19: Schéma du modèle de file d'attente M/M/1 52
Figure 20: Schéma du modèle d file d'attente M/M/s/K 54
Figure 21: Schéma du modèle avec N files d'attente avec N serveurs parallèles 56
Figure 22: Schéma du modèle avec 1 file d'attente avec N serveurs parallèles 57
Figure 23: Illustration de l'attaque par violation gestion d'authentification et session 61
Figure 24 : Illustration de l'attaque par exploitation des failles XSS 62
Figure 25: Illustration de l'attaque par CSRF 65
Figure 26: Architecture 3-Tiers, Client/Serveur 74
Figure 27: Schéma de présentation du modèle MVC 75
Figure 28: Formulaires d'authentification 80
Figure 29: Formulaires d'inscription d'un candidat électeur 81
Figure 30: Interface d'affichage du rapport individuel d'un agent recenseur 82
Figure 31: Interface d'affichage du rapport du centre d'inscription 83
Figure 32: Interface d'affichage du rapport de la CECI 84
Figure 33: Interface d'affichage du rapport de la CEPI 85
Figure 34: Interface d'affichage du rapport de la CENI 86

Liste des tableaux

Table 1: Identification des acteurs et cas d'utilisation 22
Table 2: Description du cas d'utilisation « Authentification » 24
Table 3: Description du cas d'utilisation « Inscrire un candidat électeur » 24
Table 4: Description du cas d'utilisation « Vérifier l'inscription d'un candidat électeur » 25
Table 5: Description du cas d'utilisation « Afficher les détails d'un candidat électeur » 26
Table 6: Description du cas d'utilisation « Mettre à jour un candidat électeur » 27
Table 7: Description du cas d'utilisation « Afficher le rapport individuel des inscriptions » 27
Table 8: Description du cas d'utilisation « Afficher la liste des 5 dernières inscriptions » 28
Table 9: Description du cas d'utilisation « Gestion des bureaux d'inscription » 29
Table 10: Description du cas d'utilisation « Afficher le rapport des inscriptions du centre d'inscription » 29
Table 11: Description du cas d'utilisation « Gestion des centres d'inscription » 30
Table 12: Description du cas d'utilisation « Afficher le rapport des inscriptions de la CECI » 30
Table 13: Description du cas d'utilisation « Gestion des CECIs » 31
Table 14: Description du cas d'utilisation « Afficher le rapport des inscriptions de la CEPI » 32
Table 15: Description du cas d'utilisation « Gestion des CEPIs » 32
Table 16: Description du cas d'utilisation « Afficher le rapport général des inscriptions » 33
Table 17: Comparaison des paramètres de performance 59

RESUME

De nos jours, les TIC sont devenues incontournables dans la vie quotidienne. Pour les entreprises et organisations, ces TIC ne cessent d'envahir et influencer les processus d'analyse, gestion et prise de décision. L'appropriation de ces dernières par les entreprises et organisations publiques et privées a pour but d'optimiser la production, la rentabilité ainsi que la qualité de service. Bref d'améliorer l'efficience des prestations.

La CENI dispose d'un système intégré de gestion des élections (SIGELE) utilisé au niveau du CTD pour les travaux de saisie des données, traitement et génération des listes et cartes d'électeurs. Par contre, le temps mis pour récolter les informations sur les électeurs suivis de leur saisie dans le système d'information pourrait être réduit en intégrant une application mobile qui permettrait entre autres d'inscrire les électeurs directement sans recourir aux registres d'inscription.

Le projet présenté dans cette thèse de mastère vise la mise en place d'un système automatisé de gestion et optimisation du processus de recensement pré-électoral par ~~de~~ la CENI, qui permettra aux agents de cette dernière d'inscrire les électeurs à partir de terminaux mobiles d'une part et aux autres acteurs impliqués dans le processus de recensement de suivre en temps réel son évolution d'autre part.

Mots clés : recensement, élections, file d'attente, modélisation, système, OWASP, application web, application mobile, sécurité, processus stochastiques, systèmes de gestion, systèmes d'information.

ABSTRACT

Nowadays, ICTs have become useful and master tools in daily life. For companies and organizations, they are constantly invading and influencing the processes of analysis, management and decision-making. Ownership of the latter by public and private companies and organizations aims to optimize production, profitability and quality of service.

The CENI owns an Integrated Electoral Management System (SIGELE) used at the CTD level for data entry, processing and generation of voters lists and cards. On the other hand, the time taken to gather voter information and input them in the system would be reduced by integrating a mobile application which would make it possible among others to register the voters directly without resorting to the using registration records.

The project presented in this Master's thesis aims at setting up an automated system for managing and optimizing the pre-electoral census process of the CENI, which will allow CENI agents to register voters from mobile terminals on the one hand and other actors involved in the census process to monitor its evolution in real time on the other.

Key words: census, elections, queuing, modeling, system, OWASP, web application, mobile application, security, stochastic processes, management systems, information systems.

Table des matières

Dédicaces i
Remerciements ii
Liste des sigles et abréviations iii
Liste des figures et schémas v
Liste des tableaux vi
RESUME vii
ABSTRACT vii
Table des matières viii
INTRODUCTION GENERALE 1
CHAPITRE 1 : APERCU GENERAL ET CADRE DU PROJET 3
I.1. Présentation de la Commission Electorale Nationale Indépendante 3
I.1.1. Structure de la Commission Electorale Nationale Indépendante 3
I.1.2. Missions de la Commission Electorale Nationale Indépendante 5
I.1.3. Fonctionnement de la Commission Electorale Nationale Indépendante 7
I.2. Présentation et actualité du projet 7
1.2.1. Présentation du sujet 7
1.2.2. Problématique et actualité liée au projet 8
1.2.3. Analyse et critique de l'existant 8
1.2.4. Proposition de solutions 9
1.2.5. Objectif du projet 9
1.2.6. Intérêt du sujet 10
1.2.7. Délimitation du sujet 11
1.2.8. Apports scientifiques et technologiques 11
1.2.9. Domaines d'application 11
1.2.10. Méthodologie 11
1.2.11. Modèles mathématiques 12
Conclusion 12
CHAPITRE 2 : CONCEPTION ET MODELISATION DU NOUVEAU SYSTEME DE GESTION ET OPTIMISATION DU PROCESSUS DE RECENSEMENT PRE-ELECTORAL 13
II.1. Présentation du langage de modélisation UML 13
II.1.1. L'approche objet comparée à l'approche procédurale 14
II.1.2. Les différents types de diagrammes 15
II.2. Cycle de vie d'un système d'information/logiciel 18
II.3. Le modèle de cycle de vie en V 19
II.4. Analyse et conception du nouveau système 20
II.4.1. Analyse et spécification des besoins du nouveau système 20
II.4.2. Diagramme des cas d'utilisation 21
II.4.3. Description des cas d'utilisation 24
II.5. Analyse comportementale du nouveau système 34

II.5.1. Diagrammes de séquences 34
II.5.2. Diagrammes d'activités 37
II.6. Modélisation du domaine et conception architecturale du nouveau système 41
II.6.1. Diagramme des classes 41
II.6.2. Diagramme des composants 43
II.6.3. Diagramme de déploiement 44
Conclusion 44
CHAPITRE 3 : MODELES ET METHODES D'OPTIMISATION DU PROCESSUS D'ENROLEMENT 45
III.1. Processus stochastiques et chaines de Markov a temps continu 45
III.1.1. Hypothèses de continuité et de dérivabilité des $\boldsymbol{pi, j(t)}$ 46
III.1.2. Equations de Kolmogorov 47
III.1.3. Loi d'évolution et équation d'équilibre d'une chaine de Markov a temps continu 47
III.2. Processus de naissance et de mort 48
III.3. Théories sur les files d'attentes et ses applications 49
III.3.1. Notation de KENDALL Georges 50
III.3.2. Types de files d'attente 51
III.3.3. Etude du régime stationnaire des files d'attente 51
III.3.4. Relation entre les paramètres de performance 52
III.4. Etude de la file M/M/1 52
III.4.1. Système de service M/M/1 52
III.4.2. Caractéristiques du système M/M/1 53
III.5. Etude des systèmes d'attente M/M/s/K 53
III.5.1. Caractéristiques du système M/M/s/K 53
III.5.2. Etude du système M/M/s/K 54
III.6. Système d'attente M/G/1 54
III.7. Etude du comportement de notre système à l'aide des files d'attente 55
III.7.1. Modèle avec files séparées (N files d'attente et N serveurs) 56
III.7.2. Modèle avec une file distribuée ou mutualisée 57
III.7.3. Comparaison des paramètres de performance des deux modèles 58
CHAPITRE 4 : ANALYSE DE LA SECURITE DU NOUVEAU SYSTEME AUTOMATISE DE GESTION ET OPTIMISATION DU PROCESSUS DE RECENSEMENT 60
IV.1. OWASP Top 10 2013 Risques de Sécurité des Applications web 60
IV.2. Pare-feu applicatif web : NAXSI pour NGINX 67
IV.3. SSL et certificats SSL 69
IV.4. Présentation de Spring Security 71
CHAPITRE 5 : IMPLEMENTATION DU NOUVEAU SYSTEME DE GESTION ET OPTIMISATION DU PROCESSUS DE RECENSEMENT PRE-ELECTORAL. 73
V.1. Design pattern Modèle-Vue-Contrôleur (MVC) 74
V.2. Plateformes, langages et technologies utilisés 76
V.3. Outils de développement 78

V.4. Serveurs d'application et proxy 79
V.5. Présentation de quelques fonctionnalités du système réalisé 79
Conclusion 86
CONCLUSIONS GENERALES 87
RECOMMANDATIONS 88
RÉFÉRENCES BIBLIOGRAPHIQUES ET WEBOGRAPHIQUES 89

INTRODUCTION GENERALE

Le monde évolue, les TIC ne cessent d'envahir et influencer notre quotidien. C'est ainsi qu'on assiste à l'intégration et l'appropriation de ces dernières dans la quasi-totalité des secteurs d'activités des entreprises et organisations publiques et privées avec un seul but - optimiser la productivité, la rentabilité et la qualité de service de ces dernières.

De nos jours, certaines entreprises locales, telle que Econet-Leo, Viettel, ISTEEBU, … ont adopté les systèmes automatisés de collecte de données sur téléphone mobile et ceux-ci leur ont permis de suivre de près, et sur une période et durée déterminées, l'évolution du processus de collecte, mais aussi connaitre, à chaque étape, les résultats provisoires ou définitifs y relatifs.

L'objectif de notre projet présenté dans cette thèse de mastère est la conception et l'implémentation d'un système automatisé de gestion et optimisation du processus de recensement pré-électoral par et pour la CENI. C'est un système qui permettra aux agents de la CENI d'inscrire les électeurs à partir de terminaux mobiles et aux autres acteurs impliqués dans le processus de recensement et de suivre en temps réel son évolution. Ainsi, la CENI bénéficiera d'un système automatisé de gestion du processus de recensement, qui lui permettra d'améliorer ses prestations en optimisant son rendement par rapport à l'approche actuelle de travail. En fin, le produit de notre travail facilitera le processus de recensement des électeurs et pourra réduire considérablement le temps d'élaboration du fichier électoral.

Pour mener à bout notre objectif et atteindre notre but, nous avons subdivisé notre travail en cinq chapitres.

Le premier chapitre présente l'entreprise hôte faisant objet et matière de notre projet, puis brosse les grandes lignes du projet en montrant l'objectif global et les objectifs spécifiques, la problématique, les solutions proposées et en fin les avantages et apports scientifiques et technologiques associés au projet qui sont utiles à la société.

Le deuxième chapitre traite les différentes étapes du cycle de vie du logiciel par confrontation ou application de ces dernières à notre projet. Ce chapitre aide dans la compréhension du domaine de notre projet ou existant en passant par l'analyse et la spécification des besoins fonctionnels et non-fonctionnels, la modélisation des structures

de données du domaine, la conception architecturale du nouveau système et enfin la modélisation de la dynamique du système à l'aide des diagrammes comportementaux.

Le troisième chapitre va nous permettre d'étudier et analyser le fonctionnement du nouveau système à l'aide des modèles mathématiques appropriés. Grâce à la modélisation basée sur les processus stochastiques et particulièrement les files d'attentes, nous allons pouvoir étudier et analyser le fonctionnement de notre système vis-à-vis des flux de candidats électeurs qui viendront se faire inscrire. L'étude et l'analyse du fonctionnement du nouveau système peuvent nous aider à dégager un principe global d'inscription et enrôlement d'électeurs au Burundi, et qui, en définitif peut déboucher sur la modélisation d'un système automatisé de gestion typique adapté au contexte socioculturel, comportemental et traditionnel des burundais et partout d'une population homogène quelconque.

Le quatrième chapitre parle des principes et méthodes de sécurité des systèmes d'information à base des spécifications de l'OWASP. Une étude de ces spécifications permet notamment de découvrir et d'analyser les éventuelles failles et vulnérabilités d'un système et nous en servir dans la mise en place d'une politique de sécurité de notre nouveau **« système automatisé de gestion et optimisation du processus de recensement.** »

Le cinquième chapitre et dernier est consacré à la réalisation de notre nouveau système. La présentation des concepts généraux des applications d'entreprise, les technologies et plateformes à utiliser, les langages de programmation, les systèmes de gestion de base de données et d'autres spécifications techniques et technologiques associées y sont décrites avant d'aborder et terminer par l'implémentation du nouveau système.

En fin, nous concluons notre thèse en formulant des recommandations à l'entreprise hôte et aux futurs chercheurs et terminer par la présentation des perspectives d'avenir pour notre projet.

CHAPITRE 1 : APERCU GENERAL ET CADRE DU PROJET

Dans ce chapitre nous allons présenter l'entreprise d'accueil qui est la Commission Electorale Nationale Indépendante (CENI), au sein de laquelle nous avons réalisé notre projet. Ensuite, l'analyse et critique du système existant vont rendre possible la formulation et proposition d'une nouvelle solution à la problématique relevée et contextualité : « le système automatisé de gestion et optimisation du processus de recensement » qui feront l'objet d'une étude détaillée tout au long de ce travail.

I.1. Présentation de la Commission Electorale Nationale Indépendante

I.1.1. Structure de la Commission Electorale Nationale Indépendante

La Commission Electorale Nationale Indépendante (CENI) est une organisation publique chargée d'organiser et réaliser les élections au niveau des collines, communes, provinces, et enfin au niveau national. Elle comprend cinq membres dont le Président, le Vice-président, le Commissaire chargé des Finances et de l'Administration, le Commissaire chargé des Opérations Electorales, Affaires Juridiques et Logistiques et le Commissaire Chargé de l'Education Civique et de la Communication. La structure organisationnelle de la CENI est reprise dans l'organigramme ci-dessous (Figure 1) [1].

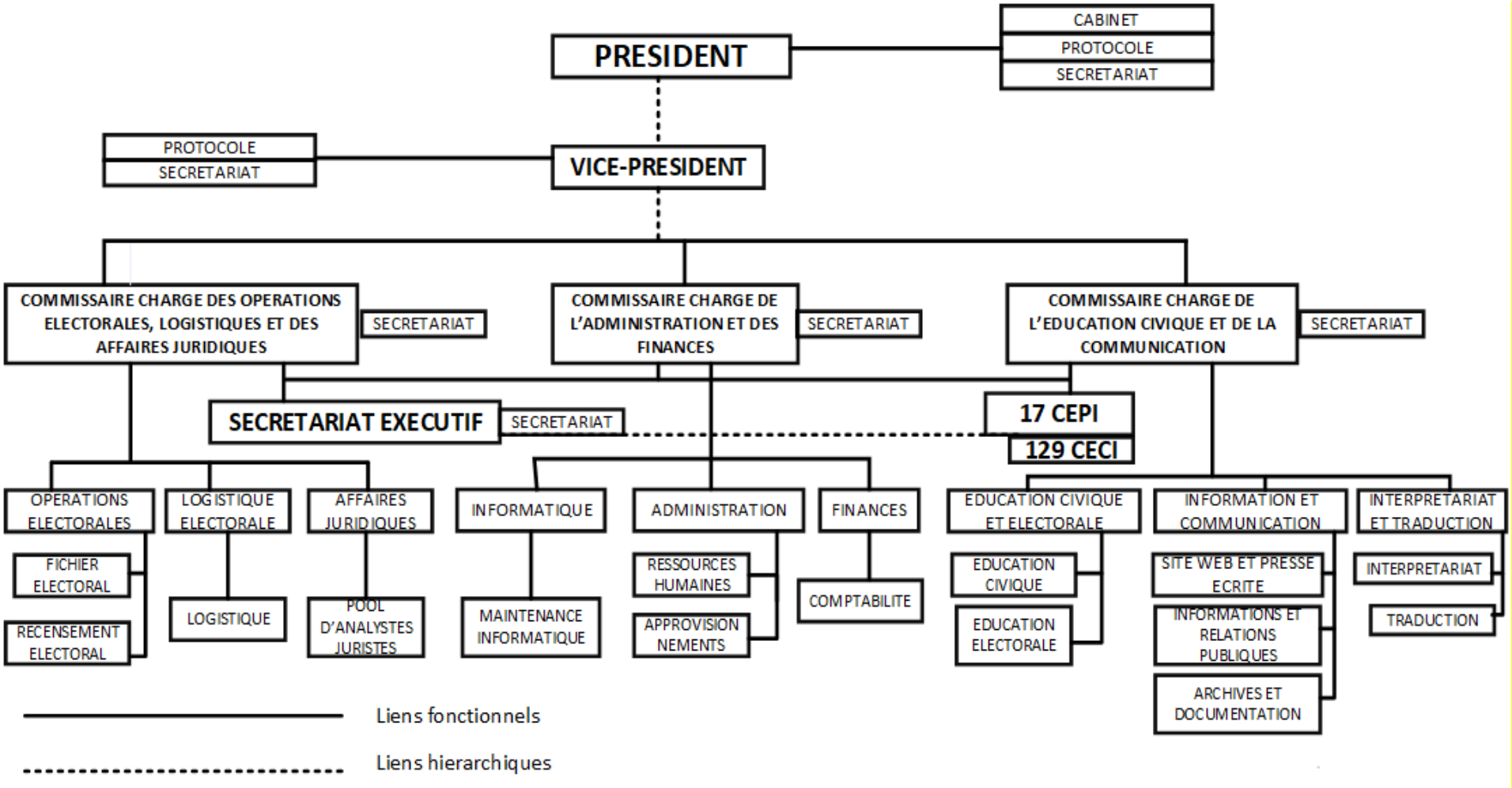


FIGURE 1: ORGANIGRAMME DE LA CENI

Au niveau du siège de la CENI, les cinq membres sont épaulés par un personnel d'appui. Chaque commissariat comprend trois services.

- Le commissariat chargé des opérations électorales, logistique électorale et affaires juridiques comprend les services opérations électorales, logistique électorale et affaires juridiques ;
- Le commissariat chargé de l'administration et des finances comprend les services des finances, administration et informatique ;
- Le commissariat chargé de l'éducation civique et de la communication comprend les services éducation civique, information et communication, interprétariat et traduction [2].

Au niveau de la province, de la commune et de la colline, la Commission Electorale Nationale Indépendante est représentée par des commissions provinciales et communales dont les membres sont nommés par la commission du niveau directement supérieur [1].

Les Commissions Electorales Provinciales Indépendantes, CEPI existent dans les 18 provinces du Burundi. Le nombre des membres de chaque CEPI est égale au nombre des communes de ladite Province auquel s'ajoute deux membres. Parmi les membres de la CEPI figure, un Président, un Vice- Président, un Chargé du matériel et ceux qui restent sont appelés membres de la CEPI [1].

I.1.2. Missions de la Commission Electorale Nationale Indépendante

Selon l'article 91 de la Constitution du Burundi et l'article 4 du décret n°100/22/ du 20 février 2009 portant organisation et fonctionnement de la Commission Electorale Nationale Indépendante,

La CENI est chargée des missions suivantes : [2]

a. Organiser les élections au niveau national, au niveau des communes et à celui des collines ;
b. Veiller à ce que ces élections soient libres, régulières et transparentes ;
c. Proclamer les résultats provisoires des élections dans un délai défini par la loi ;
d. Promulguer les arrangements, le code de conduite et les détails techniques, y compris l'emplacement des bureaux de vote et les heures auxquelles ils sont ouverts et fermés ;
e. Entendre les plaintes concernant le respect des règles électorales et y donner suite. Les décisions de la Commission sont sans appel ;

f. Veiller, en appliquant des règles appropriées, à ce que les campagnes électorales ne se déroulent pas de manière à inciter la violence ethnique ou toute autre manière contraire à la loi ;
g. Assurer le respect des dispositions de la Commission relatives à la multiethnicité et au genre et connaître des contestations à cet égard.

Ainsi donc, l'enrôlement des électeurs est l'une des étapes du cycle électoral.

Comme le montre le schéma du cycle électoral (Figure 2), l'enrôlement des électeurs consiste entre autres à faire le recensement des candidats électeurs, la saisie dans le système d'information pour produire les listes d'électeurs, la production des cartes d'électeurs, déclarations des candidatures des partis politiques et des indépendants, et enfin l'accréditation des différents observateurs [2].

Notre projet couvrira les deux premiers aspects de cette étape du cycle électoral à savoir : recensement des candidats électeurs et la saisie dans le système d'information grâce au système de gestion et optimisation du processus de recensement que nous allons concevoir et implémenter.

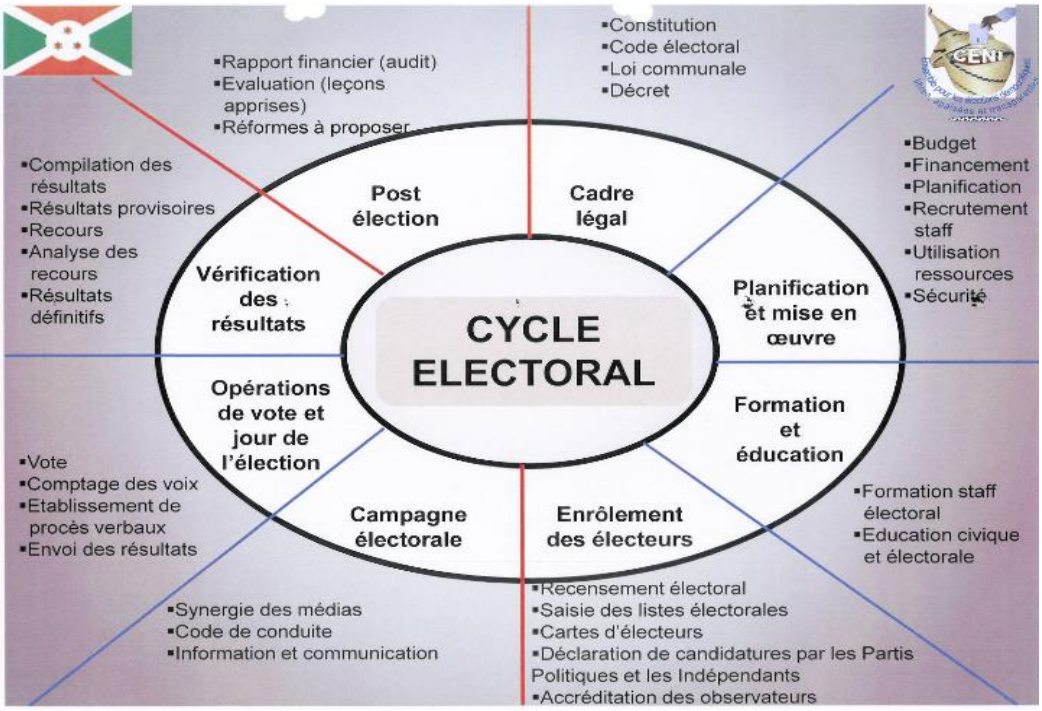


FIGURE 2: CYCLE ELECTORAL

I.1.3. Fonctionnement de la Commission Electorale Nationale Indépendante

La commission est présidée par son Président ou en cas d'empêchement par son Vice-président. Elle se réunit autant de fois que de besoin sur convocation du Président ou de son Vice-président en cas d'empêchement du Président. Toutefois, lorsque trois membres de la Commission le demandent, le Président est tenu de convoquer la réunion. La Commission décide par consensus ou à défaut à la majorité des 3/5 de ses membres. Les décisions de la CENI en matière électorale sont des arrêtés qui ont force d'exécution. La CENI planifie, organise et supervise l'exécution de toutes les activités du ressort de son domaine. Les ordres vont du siège de la CENI aux CEPI et des CEPI aux CECI [1].

Toutefois, la communication s'effectue régulièrement dans les deux sens. Différentes réunions d'échanges et d'information sont régulièrement organisées ; soit au siège de la CENI ou dans les provinces. Les décisions importantes sont prises par les membres de la CENI. Cependant, les Présidents des CEPI ont le droit de prendre des décisions d'ordre pratique liées aux cas particuliers mais qui entrent dans la ligne directrice tracée par la CENI [1].

En ce qui concerne le budget de la CENI, les ressources de la Commission proviennent des subventions inscrites annuellement au budget de l'Etat, des fonds provenant des bailleurs bilatéraux et multilatéraux, des dons et legs [1].

I.2. Présentation et actualité du projet

1.2.1. Présentation du sujet

L'une des missions confiées à la CENI est d'organiser les élections au niveau national, communal et collinaires [2]. Ainsi, l'organisation des élections suppose un recensement préalable de la population en âge de voter (électeurs potentiels), ce qui permettra à la CENI de confectionner les listes des candidats électeurs afin de s'en servir dans la planification des étapes suivantes notamment les prévisions du nombre et des lieux des bureaux de vote, la production des cartes d'électeur et des bulletins de vote, etc.

Dans ce projet, nous nous proposons de mettre en place un système de gestion automatisé qui permettra de récolter les informations des candidats électeurs de façons simple et optimale. Par conséquent, la solution logicielle à implémenter permettra de bien suivre le processus de recensement, l'analyse des données collectées et pourra contribuer dans la prise de décision par rapport aux autres étapes du cycle électoral.

1.2.2. Problématique et actualité liée au projet

La CENI comme tant d'autres organisations locales, utilise un système d'information classique (systèmes de gestion et traitement des données) dans la gestion du processus d'enrôlement des candidats électeurs. Des agents recenseurs inscrivent les candidats électeurs dans les registres d'inscriptions. Par après, ces derniers sont saisis dans un Système Intégré de Gestion des Elections (SIGELE) au niveau du CTD. Le problème est qu'on ne peut pas suivre l'évolution du processus de recensement en temps réel. En effet, on devra attendre la fin de la saisie et le traitement des données pour savoir l'effectif des personnes inscrites, les retards éventuels de la planification des étapes suivantes du cycle électoral, etc.

Des travaux d'optimisation du processus de recensement s'avèrent une nécessité. Le seul moyen à portée de main, réalisable et qui apporte une solution pérenne, c'est mettre en place un système automatisé de gestion de cette partie du cycle électoral.

1.2.3. Analyse et critique de l'existant

Dans cette section, nous passons à l'analyse et l'évaluation du système existant afin d'en dégager ses côtés forts et faibles dans le but de rechercher et proposer des solutions qui résolvent les problèmes relevés et qui satisfont aux besoins et attentes réels du bénéficiaire, la CENI et des utilisateurs. Le but de la critique de l'existant est d'établir un diagnostic précis sur les procédures utilisées, relever les anomalies, les qualités et les défauts du système existant.

Lors de cette critique, deux aspects ont été dégagés dont l'un est positif et l'autre négatif. Ces deux aspects méritent d'être soulevés étant donné que le besoin de perfectionnement sera toujours souhaité par les utilisateurs en vue d'un fonctionnement optimale du système et l'amélioration de la qualité de service et des prestations de tous les acteurs impliqués.

L'analyse du système existant nous a permis de constater et de conclure que la CENI a un système d'organisation bien défini et donc bien structuré du point de vue fonctionnel et organisationnel en matière du cycle électoral depuis le recensement d'électeurs potentiels aux scrutins proprement dits. La CENI dispose d'un centre de traitement de données (CTD) équipé d'ordinateurs en quantité suffisante utilisés dans le travail de saisi des données récoltées pendant le recensement. En plus, le « Système Intégré de Gestion des Elections - SIGELE » est utilisé dans le processus de traitement

des données saisies. Signalons en outre que ce logiciel a été développé par un expert international recruté par le PNUD pour le compte du projet PACE 2015.

Malgré cet aspect organisationnel clair et précis, l'enrôlement des candidats électeurs est effectué par des agents recenseurs qui utilisent des registres d'inscription physiques lors des inscriptions et qui sont remis à la CENI à la fin du processus de recensement. A ce niveau l'inconvénient est que le travail se fait d'abord manuellement avec un risque élevé de commettre des erreurs mais aussi doublement. En effet, les opérations commencent, puis les travaux de saisis suivent permettant ainsi de connaitre les statistiques des candidats électeurs. Hormis le double travail et l'existence d'une probabilité élevé d'erreurs, dans l'existant, il est difficile quasiment impossible de connaitre et suivre en temps réel le processus de recensement ou inscriptions d'électeurs.

De plus, le temps de saisie et traitement des données récoltées lors du processus de recensement des candidats électeurs ainsi que les dépenses que cette tache engendre sont élevés.

1.2.4. Proposition de solutions

Les fruits de l'analyse et critique faites sur le système existant nous ont conduit à l'élaboration d'une proposition de solutions, qui une fois implémentée permettrait d'optimiser le processus de recensement des candidats électeurs. Elle consistera à mettre sur pied un système automatisé de gestion et optimisation du processus de recensement pré-électoral de la CENI.

Ce système permettra aux agents recenseurs d'inscrire les candidats électeurs automatiquement dans une base de données centralisée sans pour autant attendre la fin du processus de recensement pour effectuer la saisie. De la même façon, les membres des CECI, des CEPI, de la CENI et les autres acteurs impliqués pourront suivre en temps réel l'évolution du processus.

1.2.5. Objectif du projet

Objectif global

L'objectif global de notre projet est de mettre en place un système automatisé de gestion et optimisation du processus de recensement pré-électoral de la CENI qui permettra de suivre en temps réel le processus de recensement, d'analyser les données sur les électeurs selon des critères spécifiques définis afin de faciliter et simplifier la

prise de décision par rapport à la suite à réserver au reste de l'ensemble du cycle électoral.

Objectifs spécifiques

Les objectifs spécifiques du présent projet sont entre autres :

- Analyser et concevoir les processus métier du système automatisé de gestion de recensement ;
- Concevoir et développer un module d'application web de gestion du processus de recensement ;
- Concevoir et développer un module d'application mobile de collecte d'inscription des électeurs ;
- Mettre en place des méthodes et algorithmes de sécurité du système automatisé de gestion du processus de recensement ;
- Intégrer de l'application mobile de collecte de données avec l'application web de gestion de données des électeurs ;
- Accompagner les utilisateurs des téléphones intelligents dans l'exploitation du système automatisé de gestion et optimiser le processus de recensement.
- Créer une base de données pour la sauvegarde et le backup des données des électeurs.

1.2.6. Intérêt du sujet

Ce travail présente à la fois un intérêt personnel, pour la CENI et pour le pays en général. Il m'a permis d'améliorer mon bagage intellectuel en appliquant les connaissances théoriques et pratiques acquises au cours de ma formation universitaire en les confrontant à un sujet réel d'actualité tel le recensement des électeurs avec téléphone mobile.

Ainsi, ça m'a permis aussi d'acquérir de l'expérience dans le domaine de la conception des applications d'entreprise ainsi que leur réalisation à l'aide des technologies de pointe.

En outre, la CENI bénéficiera d'un système automatisé de gestion du processus de recensement, qui lui permettra d'optimiser son rendement par rapport à l'approche actuelle de travail. La qualité de service va augmenter et ceci aura aussi un impact positif sur le processus électoral du fait que l'inscription des électeurs sera faite dans les délais.

Ainsi les données saisies sur terrain avec l'application mobile seront envoyées au serveur et sauvegardées dans la base de données en un seul clic.

D'un autre côté, le gouvernement va aussi en bénéficier du fait que l'implémentation d'un tel système rentre dans le contexte de la mise en place d'un cyber gouvernement ou e-Gouvernement.

En fin, le produit de mon travail facilitera le processus de recensement des électeurs et permettra d'améliorer le rendement en accélérant le processus d'élaboration du fichier électoral.

1.2.7. Délimitation du sujet

Le champ d'étude de notre sujet s'est limité à la mise en place d'un système automatisé pour l'optimisation du processus de recensement des électeurs. Ceci concerne la gestion des agents recenseurs, encadreurs, superviseurs, des centres d'inscription, l'inscription des candidats électeurs et tout autre acteur et/ou activité connexe au processus de recensement.

1.2.8. Apports scientifiques et technologiques

Les apports scientifiques et technologiques touchent des aspects transversaux et donc diversifiés. Les résultats de ce travail peuvent être utiles et exploités dans la conception et implémentation d'applications mobiles, intégration des services web et applications mobiles, design et implémentation des systèmes distribués, développement d'applications multicouches, techniques d'intégration des systèmes d'information, implémentation du SOA (Service Oriented Architecture) ; etc.

1.2.9. Domaines d'application

Les domaines d'application des résultats de ce projet sont entre autres : enquête, analyse et gestion de données statistiques, systèmes de gestion et optimisation du processus de recensement, organisation et gestion du processus électoral, systèmes d'aide à la prise de décision/systèmes experts et autres systèmes d'informations dont les processus métiers reflètent les mêmes caractéristiques.

1.2.10. Méthodologie

La mise en place de systèmes d'information obéit a pas mal de règles et méthodes, c'est ainsi qu'une collecte d'informations et une analyse préliminaire doivent être faites,

et par la suite la conception, l'implémentation, le déploiement et enfin la maintenance du système.

Lors de la récolte des informations nécessaires pour la conception et développement du système, nous avons consulté les documentations et archives ainsi que le site web de la CENI qui contiennent pas mal d'informations concernant cette institution. En plus, des descentes sur terrain ont été faites ou cours desquelles je me suis entretenu avec le chargé du service des opérations électorales, du recensement électoral et du fichier électoral qui m'a fourni tous les détails sur le processus de récolte des données des électeurs et la mise en place du fichier électoral en général.

D'autres documents et publications en rapport avec les enquêtes statistiques, les recensements, etc. ont été consultés pour s'enquérir des informations supplémentaires qui ont grandement contribués dans la compréhension de l'existant afin de mettre en place un nouveau système de gestion du fichier électoral.

1.2.11. Modèles mathématiques

Les modèles mathématiques nous ont permis de dimensionner notre système, de pouvoir étudier et analyser son fonctionnement à l'aide des outils scientifiques appropriés et de voir son comportement vis-à-vis des flux des personnes qui viendront se faire inscrire.

Ainsi notre système a été assimilé à une file d'attente dont les clients sont les personnes qui viennent se faire inscrire et les serveurs, les agents recenseurs. Par la suite, nous avons procédé aux calculs probabilistes afin de pouvoir estimer les différents paramètres de performance de notre système.

Conclusion

Ce chapitre introductif nous a permis de détailler le cadre général de notre projet. Et ce en présentant l'entreprise d'accueil, le système que nous allons concevoir et développer, ainsi que les objectifs visés.

Dans le chapitre qui suit nous allons passer à la phase de conception de notre système automatisé de gestion et optimisation du processus de recensement pré-électoral.

CHAPITRE 2 : CONCEPTION ET MODELISATION DU NOUVEAU SYSTEME DE GESTION ET OPTIMISATION DU PROCESSUS DE RECENSEMENT PRE-ELECTORAL

On assiste aujourd'hui aux développements massifs des systèmes d'informations de gestion dans la majorité des entreprises. Toutefois, quels que soient les domaines d'application et les objectifs à atteindre, force est de constater que la mise en œuvre de ces systèmes relève souvent d'une démarche pragmatique sans fondement conceptuel [8].

La construction d'un outil pour l'analyse et la gestion d'information de l'entreprise nécessite de se forger une représentation de la réalité étudiée, en décrivant les objets que l'on souhaite retenir, leurs caractéristiques ainsi que leurs relations. Par ailleurs, il s'agit aussi d'identifier les fonctionnalités du système, aussi bien en termes de gestion et d'affichage, qu'en termes de traitement et d'analyse de cette information. Les différents acteurs concernés et les interactions entre acteurs et systèmes doivent aussi être identifiés [8].

Actuellement, l'approche UML semble remplacer les méthodes classiques utilisées en matière de conception des systèmes d'information d'entreprise. En effet, UML permet la spécification des structures statiques et dynamiques des systèmes d'information et apparaît donc comme une approche complète dans le processus de conception et de réalisation des systèmes d'information.

II.1. Présentation du langage de modélisation UML

UML est un langage de modélisation conçu pour construire, visualiser, et spécifier les systèmes d'information (BOOCH et al, 1998), (MORLEY et al, 2000). C'est un langage de modélisation unifié permettant de modéliser une application logicielle de façon standard dans le cadre de conception orienté objet. La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc. [7]

UML est né de la fusion des trois méthodes qui ont influencé la modélisation objet au milieu des années 90 : OMT, Booch et OOSE. Il s'agit d'un compromis qui a été trouvé par une équipe d'experts : Grady Booch, James Rumbaugh et Ivar Jacobson. Il est à présent un standard défini par l'Object Management Group (OMG). De très nombreuses entreprises de renom ont adopté UML et participent encore aujourd'hui à son développement. Il est surtout utilisé lorsqu'on prévoit de développer des applications avec une démarche objet (développement en Java, en C++, etc.). [7]

En définitive, modéliser consiste à décrire de manière visuelle et graphique les besoins et, les solutions fonctionnelles et techniques d'un projet logiciel. UML nous aide à faire cette description de façon graphique et devient alors un excellent moyen pour « visualiser » le(s) futur(s) logiciel(s). Tout comme la construction d'une maison nécessite des plans à différents niveaux (vision extérieure, plan des différents étages, plans techniques…), la réalisation d'un logiciel ou d'un ensemble de logiciels a besoin d'un certain nombre de diagrammes. [7]

II.1.1. L'approche objet comparée à l'approche procédurale

Dans la gestion de projet, nous pouvons citer deux approches permettant de définir les besoins que sont la décomposition fonctionnelle (ou l'approche procédurale) et l'approche objet (sur laquelle est basée UML)

Pendant longtemps, de nombreux logiciels étaient conçus par l'approche fonctionnelle descendante, appelée également la décomposition fonctionnelle. Cette approche considère que le logiciel est composé d'une hiérarchie de fonctions et de données, les fonctions fournissent les services désirés et les données représentent les informations manipulées. Ayant constaté les problèmes inhérents à la décomposition fonctionnelle, de nombreux experts se sont penchés sur une approche différente. De cette réflexion est née l'approche objet, dont UML s'inspire largement. [35]

L'approche objet est une démarche qui s'organise autour de 4 principes fondamentaux suivants : itérative et incrémentale, guidée par les besoins du client et des utilisateurs, centrée sur l'architecture du logiciel, et qui décrit les actions et les informations dans une seule entité.

L'approche objet nécessite une démarche itérative et incrémentale, c'est-à-dire que le concepteur doit faire des allers-retours entre les diagrammes initiaux et, les besoins du client et des utilisateurs perçus au fur et à mesure de la conception du logiciel afin de le modifier si nécessaire. [35]

Les besoins des utilisateurs servent de fil rouge, tout au long du cycle de développement : lors de la phase d'analyse pour la clarification, l'affinage et la validation des besoins des utilisateurs, lors de la phase de conception et de réalisation pour la vérification de la prise en compte des besoins des utilisateurs, lors de la phase de test afin de garantir la satisfaction des besoins. [35]

L'architecture logicielle décrit les choix stratégiques qui déterminent en grande partie les qualités du logiciel (adaptabilité, performances, fiabilité...). On peut citer l'architecture client-serveur, en couche ou en niveaux. L'approche objet est centrée sur le diagramme de classes qui décrit aussi bien des actions que des informations dans une même entité. [35]

Les autres diagrammes nous aident à voir clair dans les besoins et dans la solution qui est à développer. Ils permettent de compléter le diagramme de classes.

II.1.2. Les différents types de diagrammes

UML compte en tout 13 diagrammes tous réalisés à partir du besoin des utilisateurs et peuvent être regroupés selon les aspects fonctionnels et l**es** aspects liés à l'architecture**.** UML modélise donc le système logiciel suivant ces deux modes de représentation. [35]

Les 4+1 vues d'un système

La conception d'un logiciel est organisée autour des aspects fonctionnels et d'architecture. Ces deux aspects sont représentés par le schéma de 4 vues, axées sur les besoins des utilisateurs (parfois intitulé des cas d'utilisation), appelé 4+1 vues. [35]

Le modèle « 4+1 » vues permet d'organiser la description du système en plusieurs vues complémentaires, chacune présentant le système selon un point de vue différent, en permettant de traiter séparément les intérêts des divers groupes d'intervenants et ainsi de mieux séparer les préoccupations fonctionnelles et les préoccupations architecturales.

Le schéma ci-dessous montre les différentes vues permettant de répondre au mieux aux besoins des utilisateurs, organisées selon les deux aspects (fonctionnels et architecture).

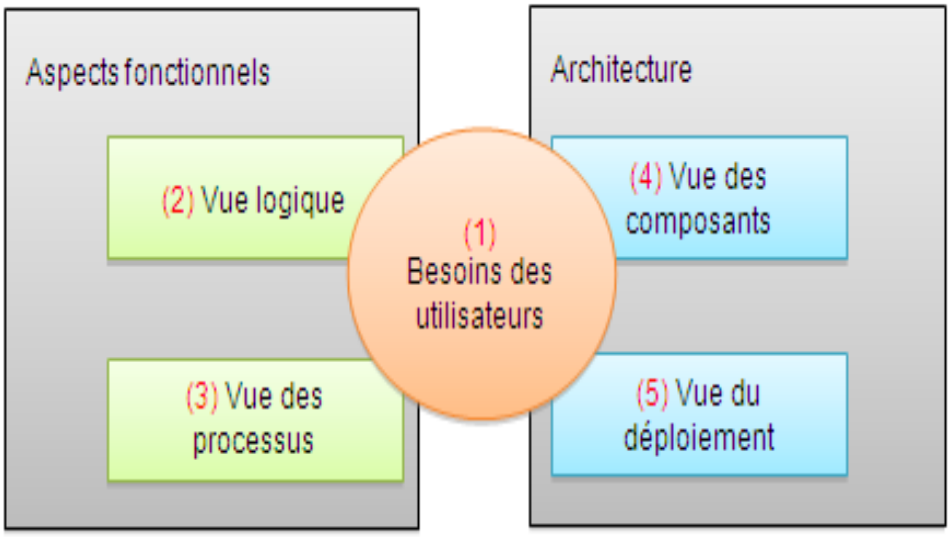


FIGURE 3: PRESENTATION DU MODELE 4 + 1 VUES

Les besoins des utilisateurs (1)

Cette partie représente le cœur de l'analyse. Elle décrit le contexte, les acteurs ou utilisateurs du projet logiciel, les fonctionnalités du logiciel mais aussi les interactions entre ces acteurs et ces fonctionnalités. Cette vue est construite en premier, juste après l'établissement du cahier des charges, pour fixer les contours du système à réaliser avec ses fonctionnalités.

Le besoin des utilisateurs peut être décrit à l'aide de deux diagrammes : [35]

- Le diagramme de packages permet de décomposer le système en catégories ou parties plus facilement observables, appelés « packages ». Cela permet également d'indiquer les acteurs qui interviennent dans chacun des packages.
- Le diagramme de cas d'utilisation représente les fonctionnalités (ou dit cas d'utilisation) nécessaires aux utilisateurs. On peut faire un diagramme de cas d'utilisation pour le logiciel entier ou pour chaque package.

Vue logique (2)

La vue logique décrit les aspects statiques et dynamiques d'un système en termes de classes, d'objets, de connexions et de communications. Elle se concentre sur l'abstraction et l'encapsulation.

Deux diagrammes peuvent être utilisés pour cette vue : [35]

- Le diagramme de classes

Dans la phase d'analyse, ce diagramme représente les entités (des informations) manipulées par les utilisateurs. Dans la phase de conception, il représente la structure objet d'un développement orienté objet.

- Le diagramme d'objets sert à illustrer les classes complexes en utilisant des exemples d'instances.

Une instance est un exemple concret de contenu d'une classe. En illustrant une partie des classes avec des exemples (grâce à un diagramme d'objets), on arrive à voir un peu plus clairement les liens nécessaires.

Vue des processus (3)

La vue des processus capte les aspects de concurrence et de synchronisation, et les décompose en flux d'exécution (processus, fil d'exécution, etc.). Elle se rapporte aux objets actifs et aux interactions.

La vue des processus s'appuie sur plusieurs diagrammes : [35]

- Le diagramme de séquence permet de décrire les différents scénarios d'utilisation du système.
- Le diagramme d'activité représente le déroulement des actions, sans utiliser les objets. En phase d'analyse, il est utilisé pour consolider les spécifications d'un cas d'utilisation.
- Le diagramme de collaboration (appelé également diagramme de communication) permet de mettre en évidence les échanges de messages entre objets. Cela nous aide à voir clair dans les actions qui sont nécessaires pour produire ces échanges de messages. Et donc de compléter, si besoin, les diagrammes de séquence et de classes.
- Le diagramme d'état-transition permet de décrire le cycle de vie des objets d'une classe.
- Le diagramme global d'interaction permet de donner une vue d'ensemble des interactions du système. Il est réalisé avec le même graphisme que le diagramme d'activité. Chaque élément du diagramme peut ensuite être détaillé à l'aide d'un diagramme de séquence ou d'un diagramme d'activité.
- Le diagramme de temps est destiné à l'analyse et la conception de systèmes ayant des contraintes temps-réel. Il s'agit là de décrire les interactions entre objets avec des contraintes temporelles fortes.

Vue des composants (4)

La vue des composants représente l'organisation statique des modules (exécutable, codes source, paquetages, etc.) dans l'environnement de développement.

Cette vue comprend deux diagrammes : [35]

- Le diagramme de structure composite décrivant un objet complexe lors de son exécution.
- Le diagramme de composants qui décrit tous les composants utiles à l'exécution du système (applications, librairies, instances de base de données, exécutables, etc.).

Vue de déploiement (5)

La vue de déploiement décrit les différentes ressources matérielles et l'implantation logicielle tenant compte de ces ressources, elle se rapporte aux nœuds physiques d'exécution et au placement des objets sur les nœuds.

Cette vue est représentée par le diagramme de déploiement, qui correspond à la description de l'environnement d'exécution du système (matériel, réseau…) et de la façon dont les composants y sont installés. [35]

II.2. Cycle de vie d'un système d'information/logiciel

Le processus de construction et de développement d'un système est un ensemble d'activité, de méthodes, de pratiques(bonnes), de fournitures et d'outils automatisés que les acteurs utilisent pour développer(produire) et maintenir un système d'information automatisée ou un logiciel [34].

Le cycle de vie d'un système logiciel se définit comme étant l'ensemble des étapes ou des phases qui interviennent dans le développement d'un système logiciel de sa conception à sa disparition [34].

En règle générale, les étapes du cycle de vie d'un système logiciel sont les suivantes : [34]

- *Définition des objectifs* : c'est dans cette étape que l'on définit la finalité du projet.
- *Analyse des besoins et faisabilité* : c'est ici que l'on réalise l'enquête, que l'on se rapproche du demandeur (client) pour recueillir ses besoins. Une fois que l'on a recueilli ces besoins, on réalise une étude de la faisabilité de ces besoins.

- *Conception générale* : ici, on élabore les spécifications de l'architecture générale du logiciel.
- *Conception détaillée* : dans cette étape, on spécifie de façon détaillée chaque sous-ensemble du logiciel que l'on va construire.
- *Programmation* : ici, on implémente les fonctionnalités définies pendant la conception.
- *Tests unitaires* : il s'agit ici de vérifier individuellement chaque sous-ensemble du logiciel que l'on veut construire.
- *Tests d'intégration* : lors de cette étape, les différents sous-ensembles sont mis ensemble pour former le logiciel complet. On vérifie ainsi les interactions entre les sous-ensembles.
- *Test d'acceptation* : on vérifie si le système répond aux besoins fonctionnels/non-fonctionnels.
- *Maintenance* : elle comporte toutes les corrections et toutes les évolutions sur le logiciel.

II.3. Le modèle de cycle de vie en V

La méthodologie adoptée dans l'analyse et la conception des systèmes représente un choix stratégique pour bien mener à terme les projets tout en respectant les délais annoncés au client et avec la qualité demandée. Pour la conception, le développement et la réalisation de notre système, nous avons opté pour l'application du processus de développement en V qui demeure actuellement le cycle de vie le plus connu et certainement le plus convenable aux projets dont les spécifications du système sont bien connues d'avance [34].

Son principe est qu'avec toute décomposition doit être décrite la recomposition, et que toute description d'un composant doit être accompagnée de test qui permettront de s'assurer qu'il correspond à sa description.

Ceci rend explicite la préparation des dernières phases (validation-vérification) par les premières (construction de l'application) et on sait progressivement si on s'approche de ce que le client désire.

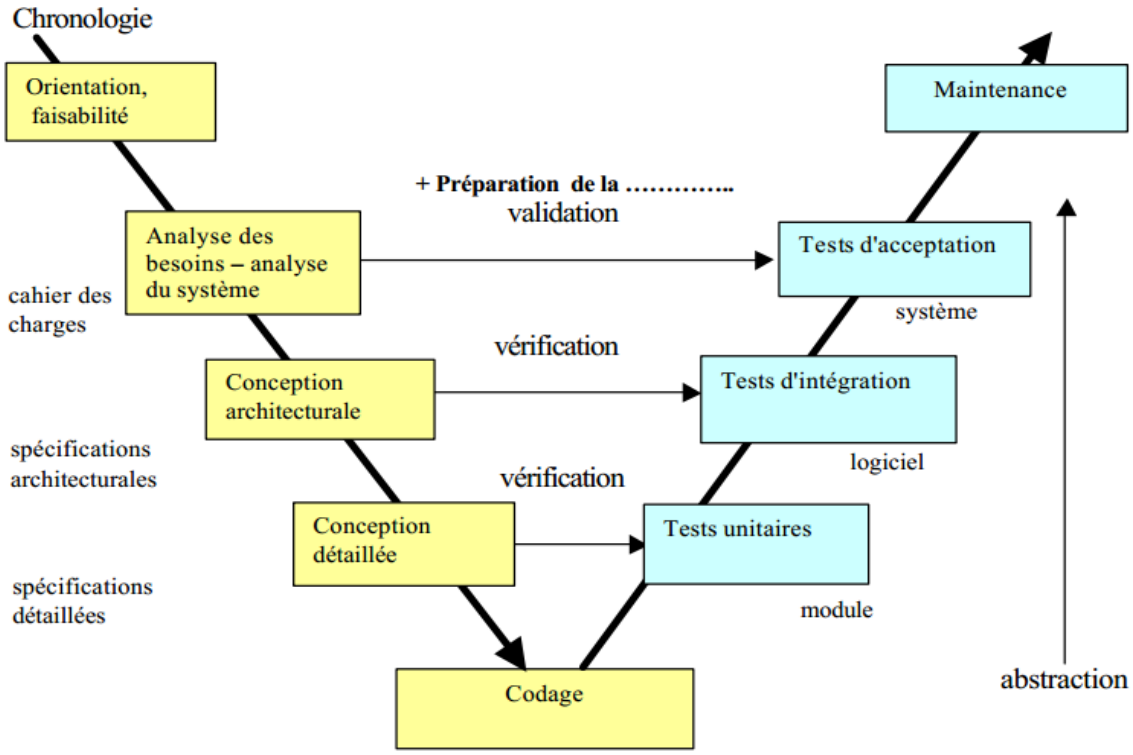


FIGURE 4: LE CYCLE DE VIE DU LOGICIEL, LE CYCLE EN V

De plus le cycle en V met en évidence la nécessité d'anticiper et de préparer dans les étapes descendantes les « attendus » des futures étapes montantes : ainsi les attendus des tests de validation sont définis lors des spécifications, les attendus des tests unitaires sont définis lors de la conception, etc.

II.4. Analyse et conception du nouveau système

Avant de passer au codage de l'application, il faut tout d'abord organiser les idées, les documenter, puis organiser la réalisation en définissant les modules et les étapes de la réalisation. Cette étape consiste à créer une représentation virtuelle des réalités du domaine afin de faire ressortir les points auxquels on s'intéresse.

II.4.1. Analyse et spécification des besoins du nouveau système

La spécification de besoins constitue la phase de départ de développement d'un système d'information, elle nous permet d'identifier les besoins du système à développer.

Nous distinguons les besoins fonctionnels qui présentent les fonctionnalités attendues de notre système et les besoins non fonctionnels qui sont des contraintes techniques, imposées par l'environnement et aussi par les utilisateurs.

La description des exigences fonctionnelles du système débouche sur la détermination des besoins des utilisateurs du système. Après analyse et étude de ces exigences, nous avons identifié les besoins utilisateurs tels que l'authentification, vérification d'une inscription d'un candidat électeur, l'inscription de candidats électeurs, la consultation de liste des candidats électeurs, la consultation des détails d'un candidat électeur, la mise à jour des informations d'un candidat électeur, la consultation du rapport des inscriptions faites, etc.

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. En ce qui concerne notre application, nous avons dégagé les besoins suivants :

- *La disponibilité* **:** l'application doit être disponible pour être utilisé par n'importe quel utilisateur depuis n'importe où il est.
- *La sécurité de l'accès aux informations critiques* **:** la conception et l'implémentation de notre système doit prendre en considération la confidentialité des données des candidats électeurs mais aussi l'authentification des différents utilisateurs du système. Pour cela nous devons restreindre l'accès aux utilisateurs qui ont les privilèges suffisants pour accéder à ces informations.
- *La convivialité de l'interface graphique* **:** l'application doit fournir une interface conviviale et simple pour tout type d'utilisateur et permettre une navigation entre différentes interfaces graphiques.
- *Une solution ouverte et évoluée* **:** possibilité d'ajouter d'autres modules pour garantir la souplesse, l'évolutivité et l'ouverture de la solution.

II.4.2. Diagramme des cas d'utilisation

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il est utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel [7].

Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

Un cas d'utilisation représente un ensemble de séquences d'actions à réaliser par le système et produisant un résultat observable intéressant pour un acteur particulier représenté par des ellipses et limité par un rectangle pour représenter le système ; tandis qu'un acteur est un utilisateur qui communique et interagit avec les cas d'utilisation du système. C'est une entité ayant un comportement comme une personne, système ou une entreprise. Un système est une composante qui fixe les limites du système en relation avec les acteurs qui l'utilisent (en dehors du système) et les fonctions qu'il doit fournir (à l'intérieur du système).

TABLE 1: IDENTIFICATION DES ACTEURS ET CAS D'UTILISATION

Acteur	Cas d'utilisation
Agent recenseur	– Inscrire les candidats électeurs – Mettre à jour les informations des candidats électeurs – Rechercher un candidat électeur connaissant son numéro d'inscription ou carte nationale d'identité – Afficher les informations d'un candidat électeur connaissant son numéro d'inscription ou carte nationale d'identité – Afficher le rapport individuel d'activités
Agent encadreur	– Il peut faire le travail d'un agent recenseur le cas échéant – Superviser le travail des agents recenseurs du centre d'inscription – Afficher le rapport d'un centre d'inscription (de tous les agents recenseur du centre)
Agent CECI	– Gestion et supervision des centres d'inscription de la CECI – Affichage des rapports d'inscription de la CECI
Agent CEPI	– Gestion et supervision des CECI de la CEPI – Affichage des rapports d'inscription de la CEPI
Agent CENI	– Gestion et supervision des CEPI – Affichage des rapports des CEPI, rapport général – Affichage du tableau de bord (Dashboard) de suivi en temps réel du processus de recensement pré-électoral de la CENI.

Sur la figure ci-dessous, est représenté le diagramme des cas d'utilisation du nouveau système de gestion et optimisation du processus de recensement pré-électoral.

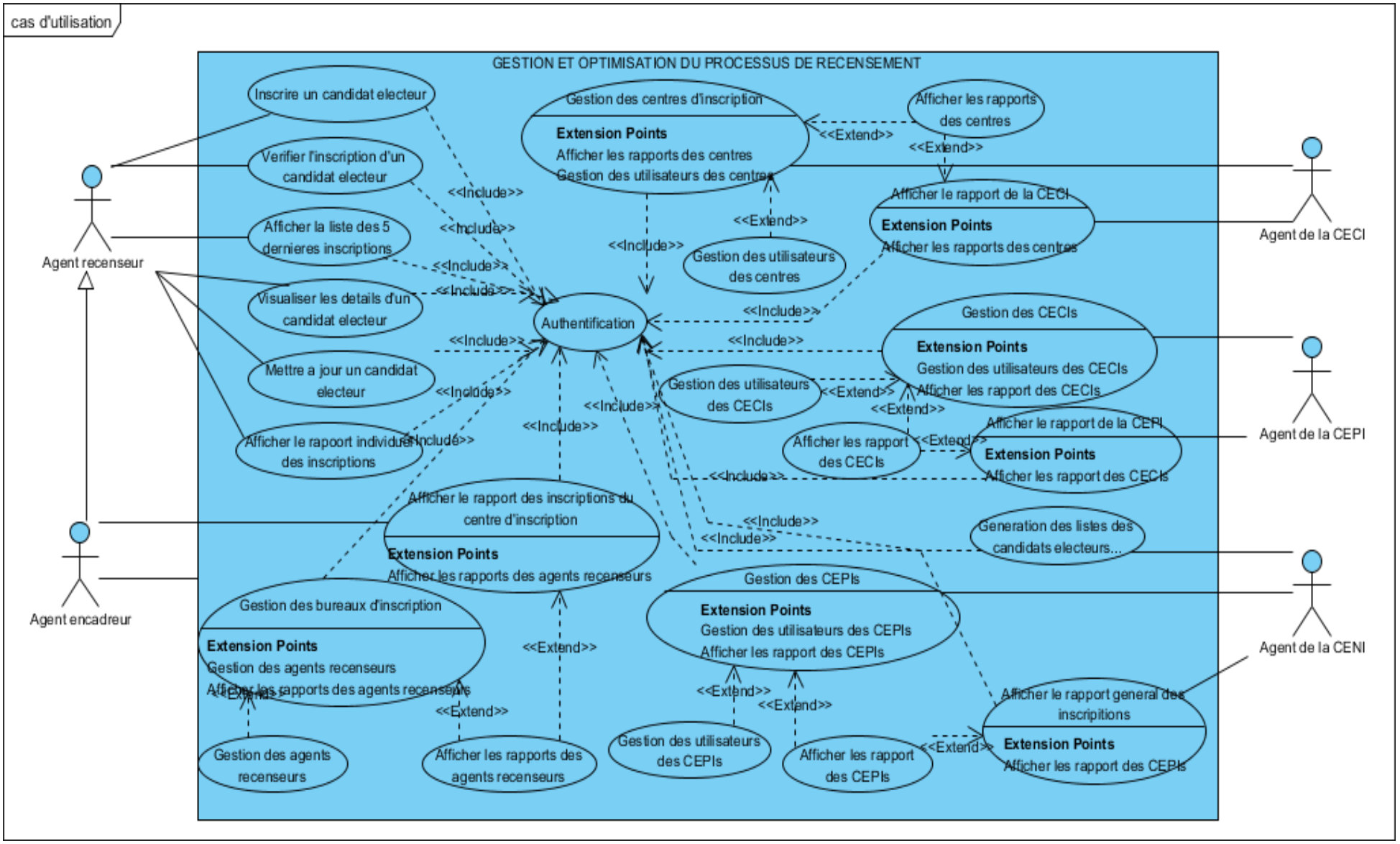


FIGURE 5: DIAGRAMME DES CAS D'UTILISATION DU NOUVEAU SYSTEME

II.4.3. Description des cas d'utilisation

TABLE 2: DESCRIPTION DU CAS D'UTILISATION « AUTHENTIFICATION »

SOMMAIRE	
Titre :	**Authentification**
But :	Se connecter au serveur pour entrer dans le système
Résumé :	L'utilisateur ouvre la page de connexion de son application mobile (ou web selon le cas), saisi le login et le mot de passe et appuie sur le bouton de connexion.
Acteur :	Agent recenseur/Agent Encadreur/Agent de la CECI/Agent de la CEPI/Agent de la CENI
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- L'application est en cours d'exécution -Affichage du formulaire de connexion	-Utilisateur authentifié et prêt à accomplir d'autres taches auxquelles il a droit.
Scénario nominal	
1. L'agent recenseur ouvre le formulaire de connexion 2. L'agent recenseur saisi le login et le mot de passe et appui sur le bouton de connexion, et est reconnu par le serveur qui lui donne accès au système. 3.Le système affiche la page d'accueil de l'application à l'utilisateur connecté	
Enchaînement alternatif	
1a. L'agent n'a pas rempli le formulaire de connexion ou il y a eu erreur de connexion ou vérification 1. Le système affiche un message d'erreur. 2. Retour à l'étape 1 du scenario nominal d'affichage du formulaire de connexion	

TABLE 3: DESCRIPTION DU CAS D'UTILISATION « INSCRIRE UN CANDIDAT ELECTEUR »

SOMMAIRE	
Titre :	**Inscrire un candidat électeur**
But :	Inscrire et enregistrer un candidat électeur dans la base de données
Résumé :	L'utilisateur ouvre le formulaire d'inscription du candidat électeur et enregistre les informations de ce dernier.
Acteur :	Agent recenseur/Agent Encadreur
DESCRIPTION DES ENCHAINEMENTS	

Pré conditions	Post conditions
- Agent recenseur est authentifié - Vérifier l'inscription du candidat électeur	-Le serveur donne la réponse de l'inscription effectuée avec succès - Le candidat électeur est inscrit, on lui remet son attestation d'inscription
Scénario nominal	
1. L'agent recenseur est connecté au système à travers son application mobile par login et un mot de passe. 2. L'agent recenseur ouvre le formulaire d'inscription et saisi les informations du candidat 3. Le système répond en affichant la liste des cinq derniers candidats électeurs inscrits par l'agent recenseur courant.	
Enchaînement alternatif	
1a. Agent n'a pas rempli correctement les champs du formulaire ou les données sont incorrectes. 1. Le système affiche un message d'erreur. 2. Retour à l'étape 2 du scénario nominal pour lancer le formulaire d'inscription.	

TABLE 4: DESCRIPTION DU CAS D'UTILISATION « VERIFIER L'INSCRIPTION D'UN CANDIDAT ELECTEUR »

SOMMAIRE	
Titre :	**Vérifier l'inscription d'un candidat électeur**
But :	Vérification de l'inscription d'un candidat électeur dans la base de données
Résumé :	L'utilisateur ouvre la page de vérification d'inscription du candidat électeur, saisi le numéro de sa carte d'identité et envoie au serveur pour vérification.
Acteur :	Agent recenseur/Agent Encadreur

DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Agent recenseur est authentifié	-Le serveur donne la réponse si le candidat électeur est inscrit ou pas.
Scénario nominal	
1. L'agent recenseur est connecté au système à travers son application mobile par login et un mot de passe. 2. L'agent recenseur ouvre la page de vérification de l'inscription, saisi le numéro de la carte d'identité du candidat et lance la vérification. 3. Le système répond en envoyant les détails d'inscription si le candidat est déjà inscrit, sinon le serveur ne donne rien comme réponse.	
Enchaînement alternatif	

1a. Agent n'a pas rempli correctement les champs du formulaire ou les données sont incorrectes. 1. Le système affiche un message d'erreur. 2. Retour à l'étape 2 du scénario nominal pour lancer la page de vérification de l'inscription du candidat.

TABLE 5: DESCRIPTION DU CAS D'UTILISATION « AFFICHER LES DETAILS D'UN CANDIDAT ELECTEUR »

SOMMAIRE	
Titre :	**Afficher les détails d'un candidat électeur**
But :	Visualiser les informations détaillées d'un candidat électeur déjà inscrit.
Résumé :	L'utilisateur ouvre la page de recherche et saisi soit le code d'électeur ou le numéro de la carte d'identité ou le numéro d'inscription. Après avoir appuyé sur le bouton rechercher, le candidat recherché s'affiche. En cliquant sur le nom du candidat, on parvient à voir les détails de ce dernier.
Acteur :	Agent recenseur/Agent Encadreur

DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Agent recenseur est authentifié	-Affichage des détails du candidat trouvé répondant aux critères spécifiés.

Scénario nominal
1. L'agent recenseur est connecté au système à travers son application mobile par login et un mot de passe. 2. L'agent recenseur ouvre la page de recherche d'un candidat électeur, saisi le numéro de la carte d'identité du candidat ou le code d'électeur et lance la recherche. 3. Le système affiche les détails du candidat électeur trouvé.
Enchaînement alternatif
1a. Agent n'a pas rempli correctement les champs du formulaire ou les données sont incorrectes. 1. Le système affiche un message d'erreur. 2. Retour à l'étape 2 du scénario nominal pour lancer la page de recherche du candidat.

TABLE 6: DESCRIPTION DU CAS D'UTILISATION « METTRE A JOUR UN CANDIDAT ELECTEUR »

SOMMAIRE	
Titre :	**Mettre à jour un candidat électeur**
But :	Mettre à jour les informations d'un candidat électeur dans la base de données
Résumé :	L'utilisateur ouvre la page de mise à jour des informations du candidat électeur, modifie ce qu'il veut et valide la modification.
Acteur :	Agent recenseur/Encadreur
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Agent recenseur est authentifié -Affichage des détails du candidat électeur	-Le serveur donne la réponse de mise à jour effectuée avec succès.
Scénario nominal	
1. L'agent recenseur est connecté au système à travers son application mobile par login et un mot de passe. 2. L'agent recenseur affiche les détails d'un candidat électeur déjà inscrit et clique sur le bouton de modification. 3. La page de mise à jour des informations du candidat électeur s'ouvre et l'agent recenseur modifie ce qu'il veut. 4. Le système répond avec le message de succès de la mise à jour effectuée avec succès.	
Enchaînement alternatif	
1a. Agent n'a pas rempli correctement les champs du formulaire ou les données sont incorrectes. 1. Le système affiche un message d'erreur. 2. Retour à l'étape 2 du scénario nominal pour lancer la page d'affichage des détails du candidat.	

TABLE 7: DESCRIPTION DU CAS D'UTILISATION « AFFICHER LE RAPPORT INDIVIDUEL DES INSCRIPTIONS »

SOMMAIRE	
Titre :	**Afficher le rapport individuel des inscriptions**
But :	Affichage du rapport individuel des inscriptions faites par jour et aussi son historique.
Résumé :	L'utilisateur ouvre la page de rapport individuel, il peut également voir l'historique des rapports des inscriptions faites par date et genre.
Acteur :	Agent recenseur/Agent Encadreur
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions

- Agent recenseur est authentifié	-Le système affiche le rapport individuel des inscriptions effectuées.
Scénario nominal	
1. L'agent recenseur est connecté au système à travers son application mobile par login et un mot de passe. 2. L'agent recenseur ouvre la page d'affichage du rapport individuel des inscriptions faites.	
Enchaînement alternatif	
Pas de scénario alternatif	

TABLE 8: DESCRIPTION DU CAS D'UTILISATION « AFFICHER LA LISTE DES 5 DERNIERES INSCRIPTIONS »

SOMMAIRE	
Titre :	**Afficher la liste des 5 dernières inscriptions**
But :	Consulter la liste des 5 derniers candidats électeurs inscrits
Résumé :	L'utilisateur ouvre la page d'affichage de la liste des candidats électeurs inscrits par lui-même, le système lui retourne la liste des 5 dernières inscriptions faites.
Acteur :	Agent recenseur/Agent Encadreur
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Agent recenseur est authentifié	-Affichage de la liste des 5 derniers candidats électeurs inscrits par l'utilisateur courant
Scénario nominal	
1. L'agent recenseur est connecté au système à travers son application mobile par login et un mot de passe. 2. L'agent recenseur affiche la liste des 5 derniers candidats électeurs inscrits par lui-même.	
Enchaînement alternatif	
Pas de scenario alternatif	

TABLE 9: DESCRIPTION DU CAS D'UTILISATION « GESTION DES BUREAUX D'INSCRIPTION »

SOMMAIRE	
Titre :	**Gestion des bureaux d'inscription**
But :	Gérer les bureaux d'inscription et les agents recenseurs
Résumé :	L'utilisateur peut ajouter, modifier et supprimer le(s) bureau(x) d'inscription et agent(s) recenseur(s) de son centre. En plus l'utilisateur peut aussi visualiser les rapports des inscriptions faites par chaque agent recenseur du centre.
Acteur :	Agent Encadreur
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Utilisateur est authentifié	-Ajout/affichage/modification/ suppression des bureaux d'inscription et des agents recenseurs -Affichage des rapports des inscriptions faites par les agents recenseurs
Scénario nominal	
1. L'agent encadreur choisit soit de gérer les bureaux, les agents recenseurs ou visualiser les rapports.	
Enchaînement alternatif	
1a. En cas d'erreur, l'utilisateur doit retourner au menu précédent afin de recommencer le processus.	

TABLE 10: DESCRIPTION DU CAS D'UTILISATION « AFFICHER LE RAPPORT DES INSCRIPTIONS DU CENTRE D'INSCRIPTION »

SOMMAIRE	
Titre :	**Afficher le rapport des inscriptions du centre d'inscription**
But :	Consulter le rapport global de tout le centre d'inscription
Résumé :	L'agent encadreur ouvre la page de rapport global du centre, il peut également consulter le rapport détaillé par genre, date, ainsi que par agent recenseur.
Acteur :	Agent Encadreur
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Utilisateur est authentifié	Affichage du rapport des inscriptions effectuées par les agents recenseurs du centre
Scénario nominal	

1. L'agent encadreur affiche le rapport global des inscriptions faites dans le centre.
Enchaînement alternatif
Pas de scenario alternatif

TABLE 11: DESCRIPTION DU CAS D'UTILISATION « GESTION DES CENTRES D'INSCRIPTION »

SOMMAIRE	
Titre :	**Gestion des centres d'inscription**
But :	Gérer les centres d'inscription et les agents recenseurs, agents encadreurs
Résumé :	L'agent de la CECI peut ajouter, modifier et supprimer le(s) centre(s) d'inscription et agent(s) recenseur(s)/encadreur(s) de la CECI. En plus l'agent de la CECI peut aussi visualiser les rapports des inscriptions faites par chaque centre de la CECI.
Acteur :	Agent de la CECI

DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Utilisateur est authentifié	-Ajout/affichage/modification/ suppression des centres d'inscription et des agents recenseurs/encadreurs -Affichage des rapports des inscriptions faites par les centres de la CECI
Scénario nominal	
1. L'agent de la CECI choisit soit de gérer les centres, les agents recenseurs/encadreurs ou visualiser les rapports.	
Enchaînement alternatif	
1a. En cas d'erreur, l'utilisateur doit retourner au menu précédent afin de recommencer le processus.	

TABLE 12: DESCRIPTION DU CAS D'UTILISATION « AFFICHER LE RAPPORT DES INSCRIPTIONS DE LA CECI »

SOMMAIRE	
Titre :	**Afficher le rapport des inscriptions de la CECI**
But :	Consulter le rapport global de la CECI
Résumé :	L'agent de la CECI ouvre la page de rapport global de la CECI, il peut également consulter le rapport détaillé par genre, date, ainsi que par centre d'inscription.
Acteur :	Agent de la CECI

DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Utilisateur est authentifié	Affichage du rapport des inscriptions effectuées par les centres de la CECI
Scénario nominal	
1. L'agent de la CECI affiche le rapport global des inscriptions faites dans les centres de la CECI.	
Enchaînement alternatif	
Pas de scenario alternatif	

TABLE 13: DESCRIPTION DU CAS D'UTILISATION « GESTION DES CECIS »

SOMMAIRE	
Titre :	**Gestion des CECIs**
But :	Gérer les CECIs et les agents de la CECI
Résumé :	L'agent de la CEPI peut ajouter, modifier et supprimer le(s) CECI(s) et agent(s) de la CECI. En plus l'agent de la CEPI peut aussi visualiser les rapports des inscriptions faites par chaque CECI de la CEPI.
Acteur :	Agent de la CEPI
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Utilisateur est authentifié	-Ajout/affichage/modification/ suppression des CECI(s) et des agents de la CECI -Affichage des rapports des inscriptions faites par les CECI(s) de la CEPI
Scénario nominal	
1. L'agent de la CEPI choisit soit de gérer les CECI(s), les agents de la CECI ou visualiser les rapports.	
Enchaînement alternatif	
1a. En cas d'erreur, l'utilisateur doit retourner au menu précédent afin de recommencer le processus.	

TABLE 14: DESCRIPTION DU CAS D'UTILISATION « AFFICHER LE RAPPORT DES INSCRIPTIONS DE LA CEPI »

SOMMAIRE	
Titre :	**Afficher le rapport des inscriptions de la CEPI**
But :	Consulter le rapport global de la CEPI
Résumé :	L'agent de la CEPI ouvre la page de rapport global de la CEPI, il peut également consulter le rapport détaillé par genre, date, ainsi que par CECI.
Acteur :	Agent de la CEPI
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Utilisateur est authentifié	Affichage du rapport des inscriptions effectuées par les CECI(s) de la CEPI
Scénario nominal	
1. L'agent de la CEPI affiche le rapport global des inscriptions faites dans les CECI(s) de la CEPI.	
Enchaînement alternatif	
Pas de scenario alternatif	

TABLE 15: DESCRIPTION DU CAS D'UTILISATION « GESTION DES CEPIS »

SOMMAIRE	
Titre :	**Gestion des CEPIs**
But :	Gérer les CEPIs et les agents de la CEPI
Résumé :	L'agent de la CENI peut ajouter, modifier et supprimer le(s) CEPI(s) et agent(s) de la CEPI. En plus l'agent de la CENI peut aussi visualiser les rapports des inscriptions faites par chaque CEPI.
Acteur :	Agent de la CENI
DESCRIPTION DES ENCHAINEMENTS	
Pré conditions	Post conditions
- Utilisateur est authentifié	-Ajout/affichage/modification/ suppression des CEPI(s) et des agents de la CEPI -Affichage des rapports des inscriptions faites par les CEPI(s)
Scénario nominal	

1. L'agent de la CENI choisit soit de gérer les CEPI(s), les agents de la CEPI ou visualiser les rapports.
Enchaînement alternatif
1a. En cas d'erreur, l'utilisateur doit retourner au menu précédent afin de recommencer le processus.

TABLE 16: DESCRIPTION DU CAS D'UTILISATION « AFFICHER LE RAPPORT GENERAL DES INSCRIPTIONS »

SOMMAIRE		
Titre :	**Afficher le rapport général des inscriptions**	
But :	Consulter le rapport global de la CENI	
Résumé :	L'agent de la CENI ouvre la page de rapport global, il peut également consulter le rapport détaillé par genre, date, ainsi que par CEPI.	
Acteur :	Agent de la CENI	
DESCRIPTION DES ENCHAINEMENTS		
Pré conditions		Post conditions
- Utilisateur est authentifié		Affichage du rapport des inscriptions effectuées par les CEPI(s)
Scénario nominal		
1. L'agent de la CENI affiche le rapport global des inscriptions faites dans les CEPI(s)		
Enchaînement alternatif		
Pas de scenario alteratif		

II.5. Analyse comportementale du nouveau système

II.5.1. Diagrammes de séquences

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages. Ils peuvent servir à illustrer un cas d'utilisation. L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme ; le temps s'écoule "de haut en bas" de cet axe [7].

1. Diagramme de séquence pour le cas d'utilisation « Authentification »

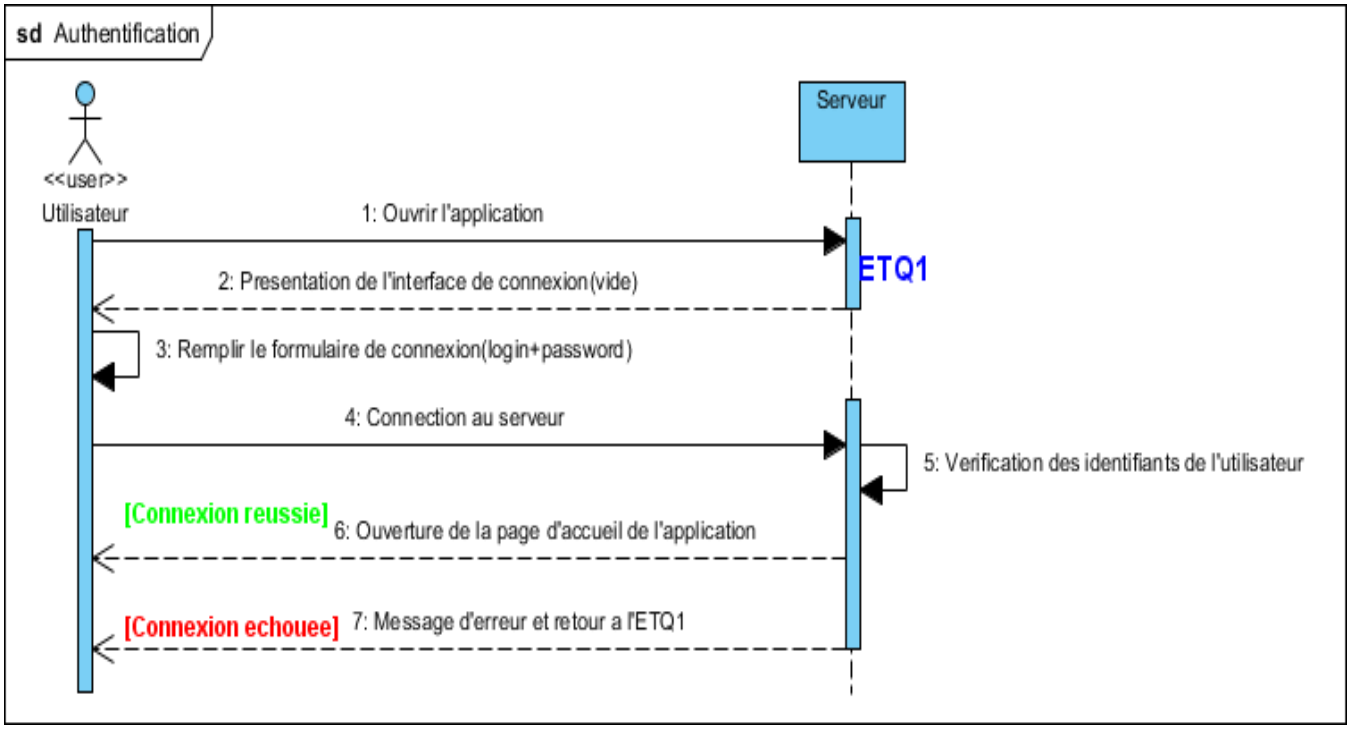


FIGURE 6 : DIAGRAMME DE SEQUENCE POUR LE CAS D'UTILISATION « AUTHENTIFICATION »

A l'ouverture l'application, la page de connexion se présente et l'utilisateur saisi ses identifiants (nom d'utilisateur et mot de passe) après quoi il tente de se connecter au serveur. Le serveur à son tour vérifie et valide les identifiants de l'utilisateur, et décide de l'authentifier ou pas en fonction du résultat de la vérification.

2. Diagramme de séquence pour le cas d'utilisation « Inscrire un candidat électeur »

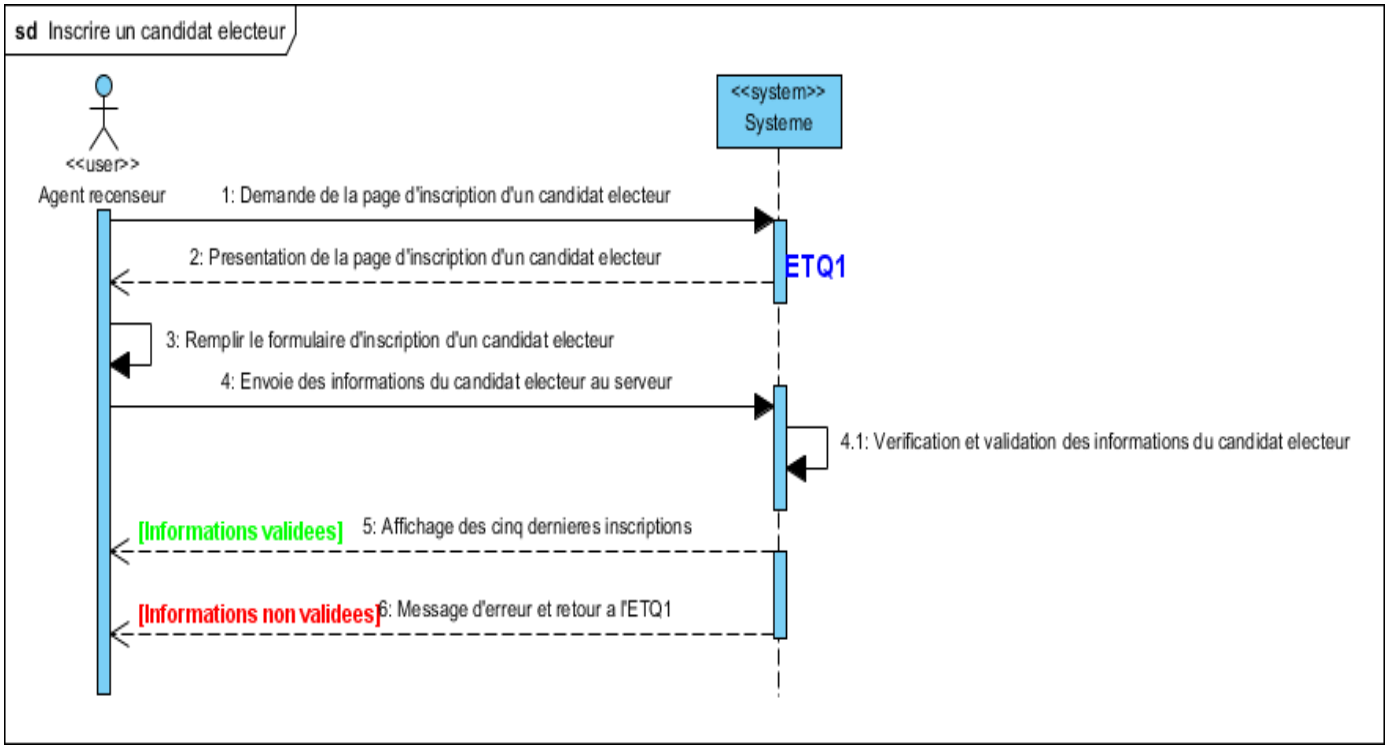


FIGURE 7: DIAGRAMME DE SEQUENCE POUR LE CAS D'UTILISATION « INSCRIRE UN CANDIDAT ELECTEUR »

L'agent recenseur doit être connecté, donc authentifié avant d'ouvrir le formulaire d'inscription. Par la suite, il va saisir les informations du candidat électeur dans le formulaire et l'envoie au serveur pour vérification et enregistrement. Si les informations envoyées sont valides, le serveur va enregistrer le candidat électeur. Sinon, il va réafficher le formulaire d'inscription du candidat avec les données et les messages d'erreurs pour guider l'agent recenseur dans la correction.

3. Diagramme de séquence pour le cas d'utilisation « Mettre à jour un candidat électeur »

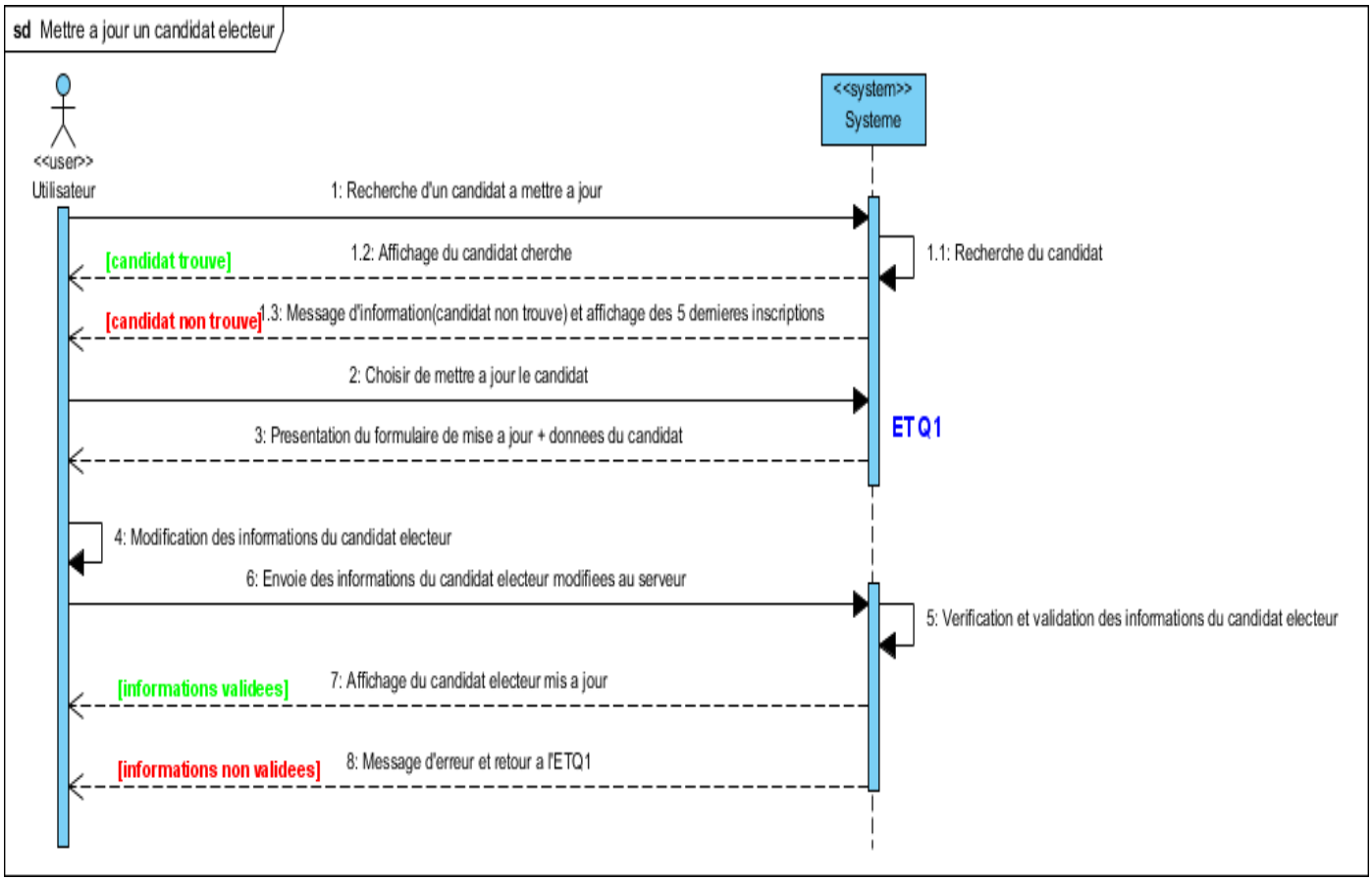


FIGURE 8: DIAGRAMME DE SEQUENCE POUR LE CAS D'UTILISATION « METTRE A JOUR UN CANDIDAT ELECTEUR »

L'agent recenseur authentifié commence par rechercher le candidat électeur à mettre à jour, et une fois trouvé choisit de modifier ses informations. Le processus continue comme pour l'inscription mais la différence est que les données sont chargées avec le formulaire, on ne doit pas saisir toutes les informations mais plutôt modifier celles qu'on veut mettre à jour.

4. Diagramme de séquence pour le cas d'utilisation « Vérifier l'inscription d'un candidat électeur »

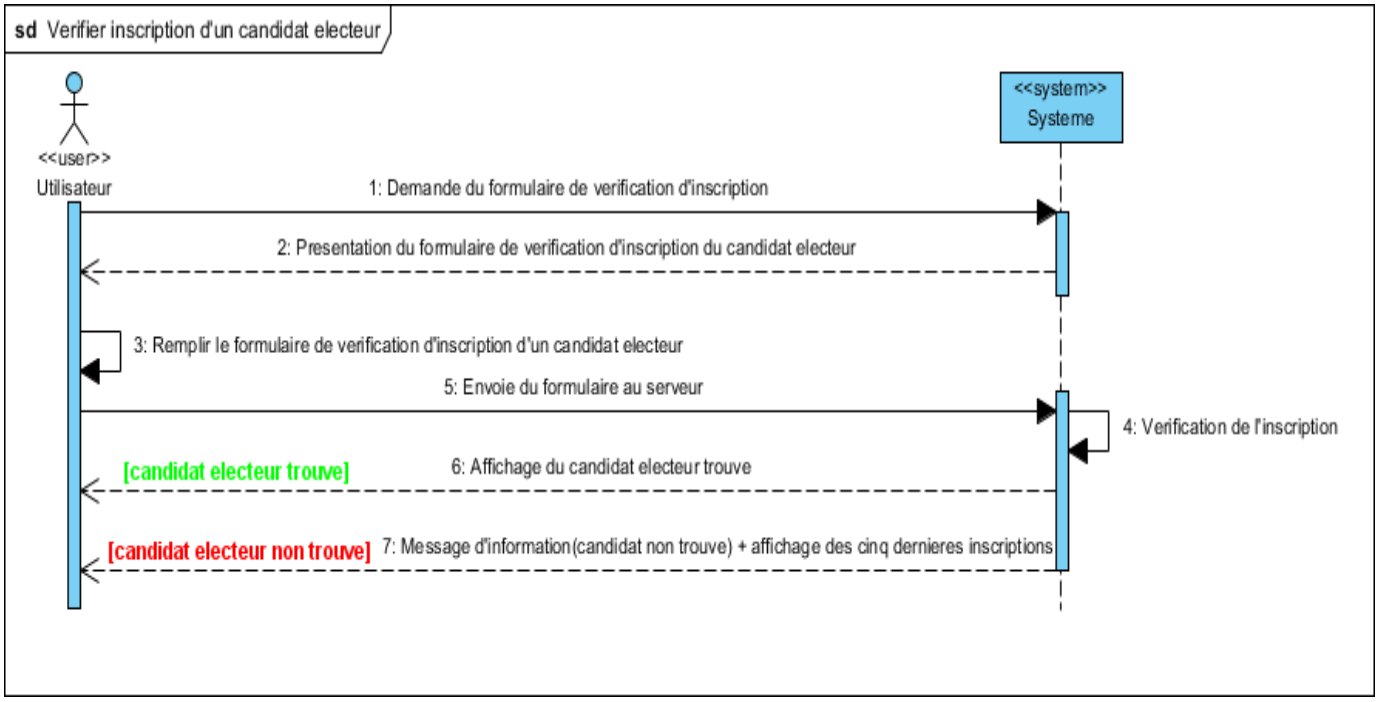


FIGURE 9: DIAGRAMME DE SEQUENCE POUR LE CAS D'UTILISATION « VERIFIER L'INSCRIPTION D'UN CANDIDAT ELECTEUR »

L'agent recenseur doit être connecté. Sur demande, le système va charger le formulaire de recherche de candidat connaissant son numéro d'inscription ou identité. L'agent recenseur saisit le numéro d'inscription ou d'identité du candidat, et l'envoie au serveur pour vérification. Une fois trouvé, le candidat est affiché sinon, le système affiche un message d'erreur comme quoi le candidat n'est pas trouvé et affiche les cinq dernières inscriptions faites par l'agent recenseur actuel.

II.5.2. Diagrammes d'activités

Une activité représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles. Les diagrammes d'activités nous permettent de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation du système [7].

Le passage d'une activité vers une autre est matérialisé par une transition. Les transitions sont déclenchées par la fin d'une activité et provoquent le début immédiat d'une autre (elles sont automatiques).

1. Diagramme d'activités pour le cas d'utilisation « Authentification »

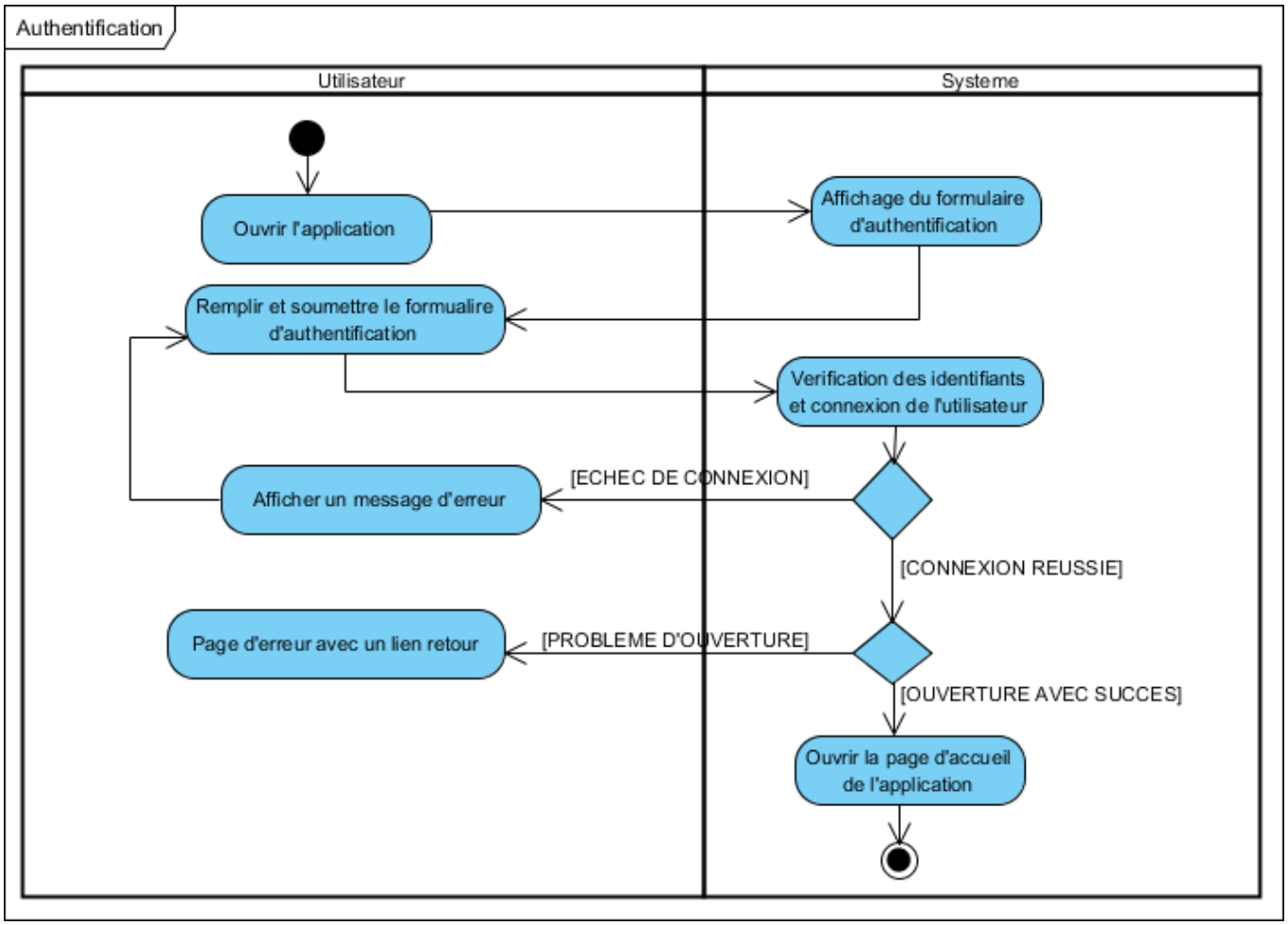


FIGURE 10: DIAGRAMME D'ACTIVITES POUR LE CAS D'UTILISATION « AUTHENTIFICATION »

Ce diagramme montre les différentes étapes pour s'authentifier au serveur d'application. C'est un enchaînement d'activités qui font interagir l'utilisateur et le système depuis l'ouverture de la page de connexion jusqu'à l'authentification réussie ou échouée.

2. Diagramme d'activités pour le cas d'utilisation « Inscrire un candidat électeur »

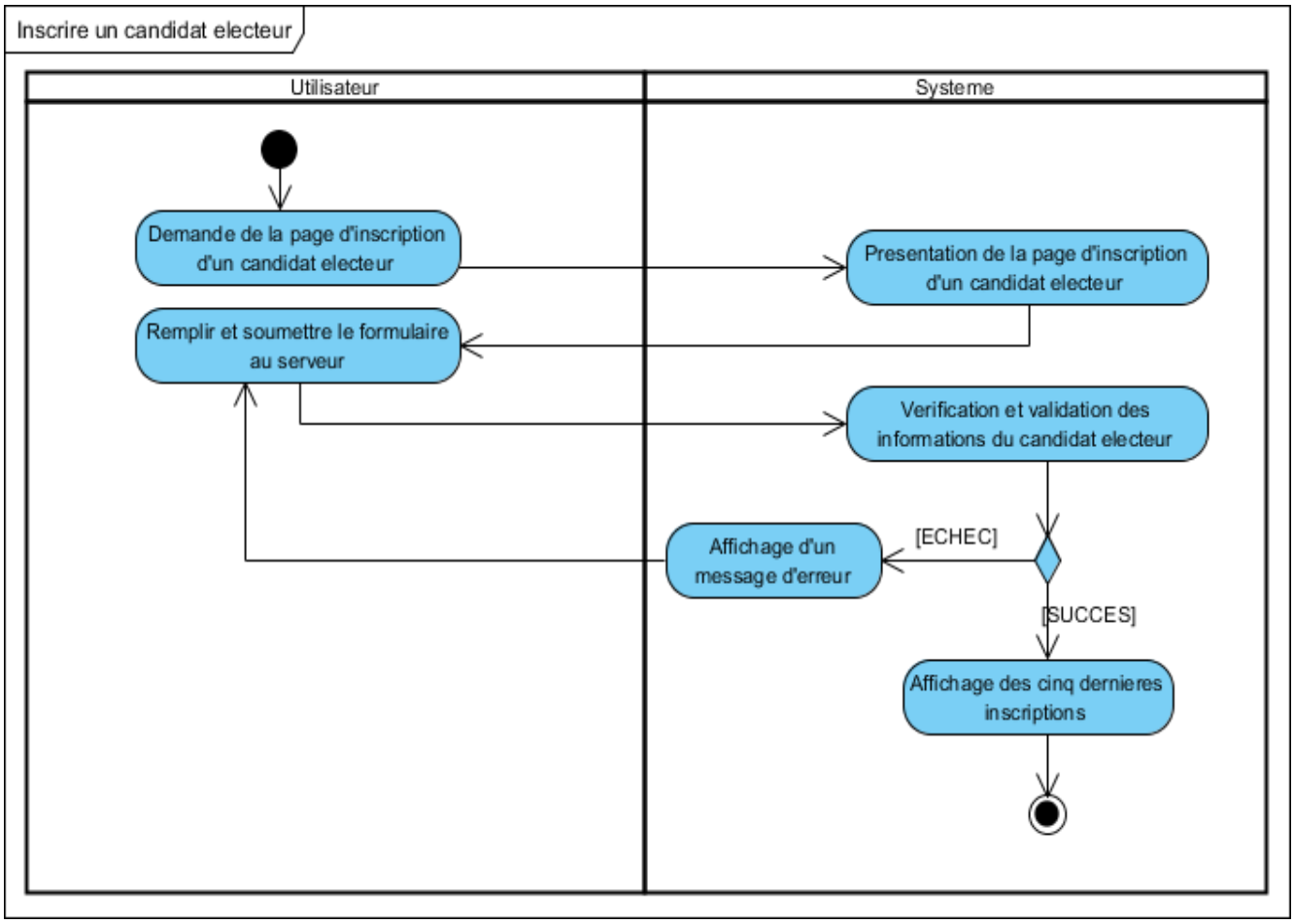


FIGURE 11: DIAGRAMME D'ACTIVITES POUR LE CAS D'UTILISATION « INSCRIRE UN CANDIDAT ELECTEUR »

Ce diagramme montre les différentes opérations qui permettent à un agent recenseur d'inscrire un candidat électeur.

3. Diagramme d'activités pour le cas d'utilisation « Mettre à jour un candidat électeur »

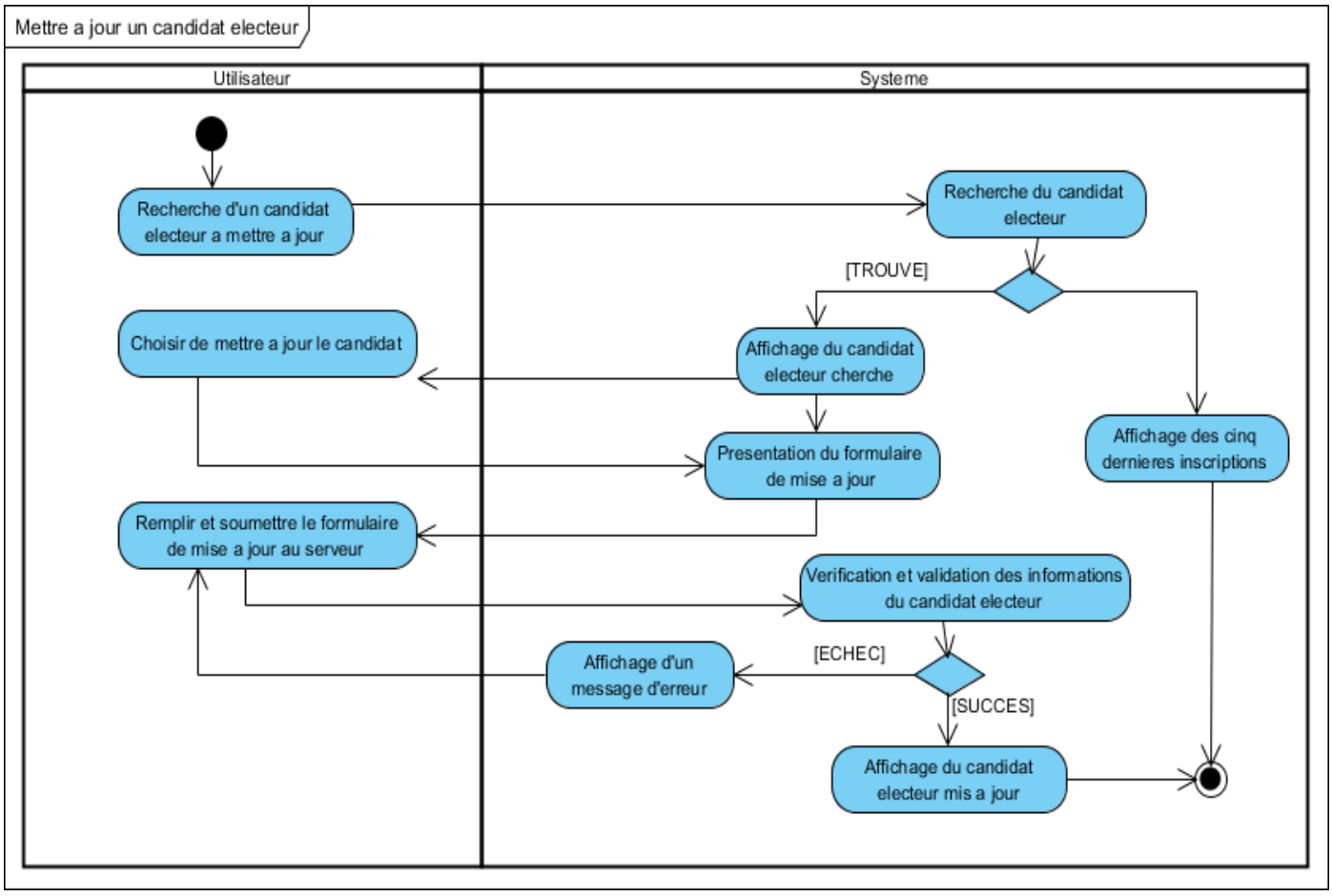


FIGURE 12: DIAGRAMME D'ACTIVITES POUR LE CAS D'UTILISATION « METTRE A JOUR UN CANDIDAT ELECTEUR »

4. Diagramme d'activités pour le cas d'utilisation « Vérifier l'inscription d'un candidat électeur »

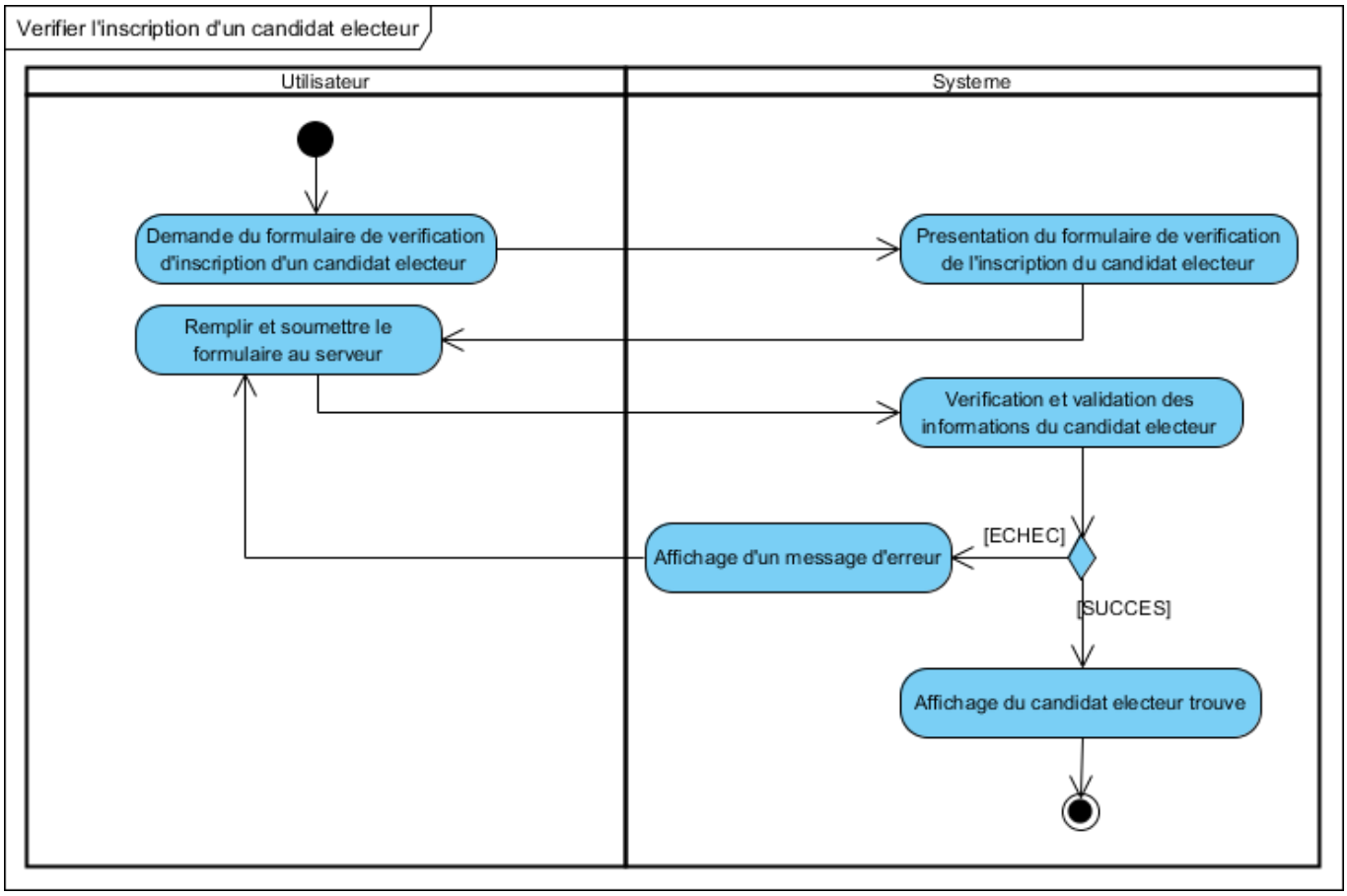


FIGURE 13: DIAGRAMME D'ACTIVITES POUR LE CAS D'UTILISATION « VERIFIER L'INSCRIPTION D'UN CANDIDAT ELECTEUR »

II.6. Modélisation du domaine et conception architecturale du nouveau système

II.6.1. Diagramme des classes

Un diagramme de classes est une collection d'éléments de modélisation statiques (classes, paquetages...), qui montre la structure d'un modèle. Un diagramme de classes fait abstraction des aspects dynamiques et temporels [7]

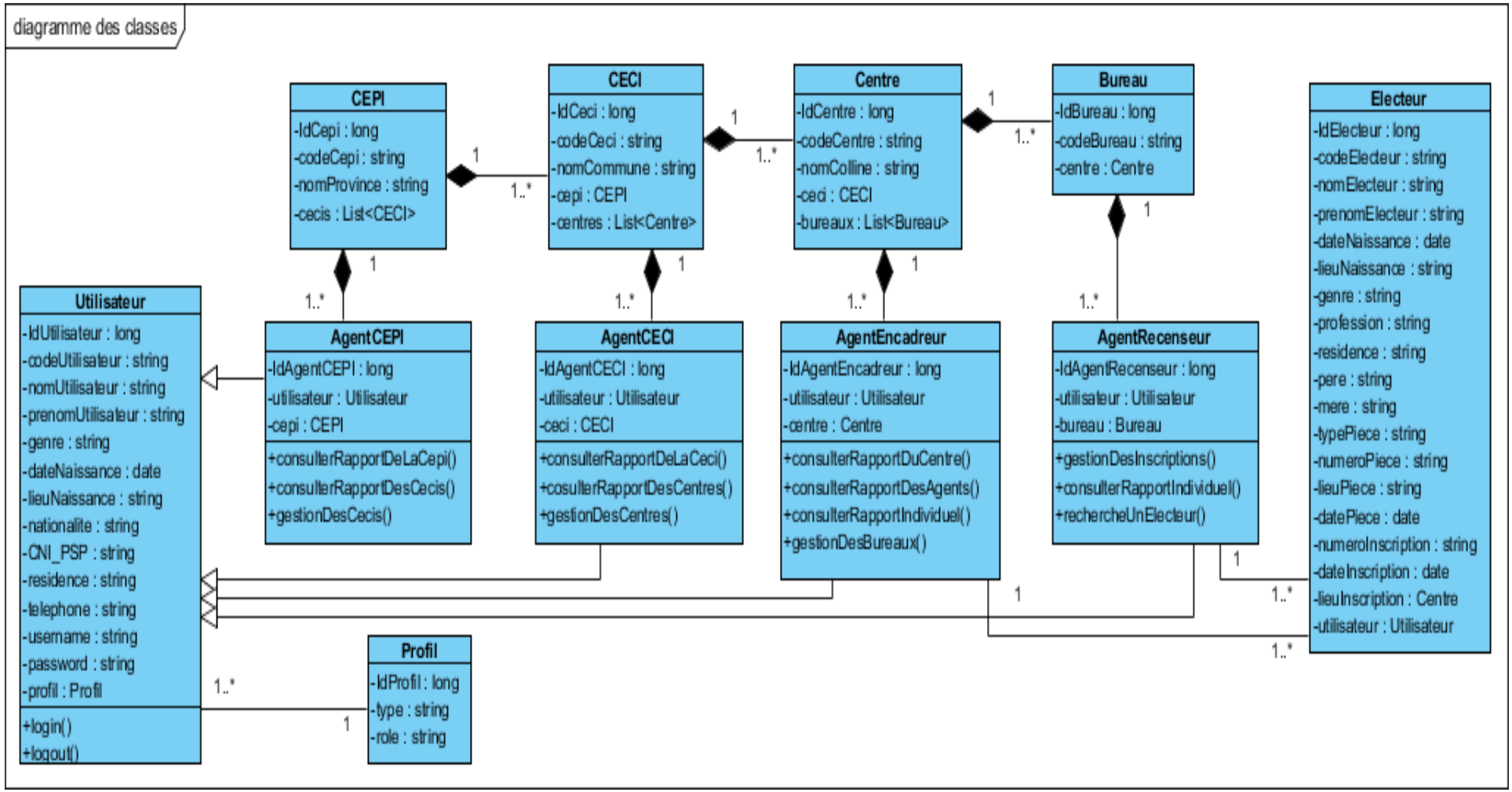


FIGURE 14: DIAGRAMME DES CLASSES

II.6.2. Diagramme des composants

Les diagrammes de composants permettent de décrire l'architecture physique et statique d'une application en termes de modules : fichiers sources, librairies, exécutables, etc [7].

Ils montrent la mise en œuvre physique des modèles de la vue logique avec l'environnement de développement. Les dépendances entre composants permettent notamment d'identifier les contraintes de compilation et de mettre en évidence la réutilisation de composants.

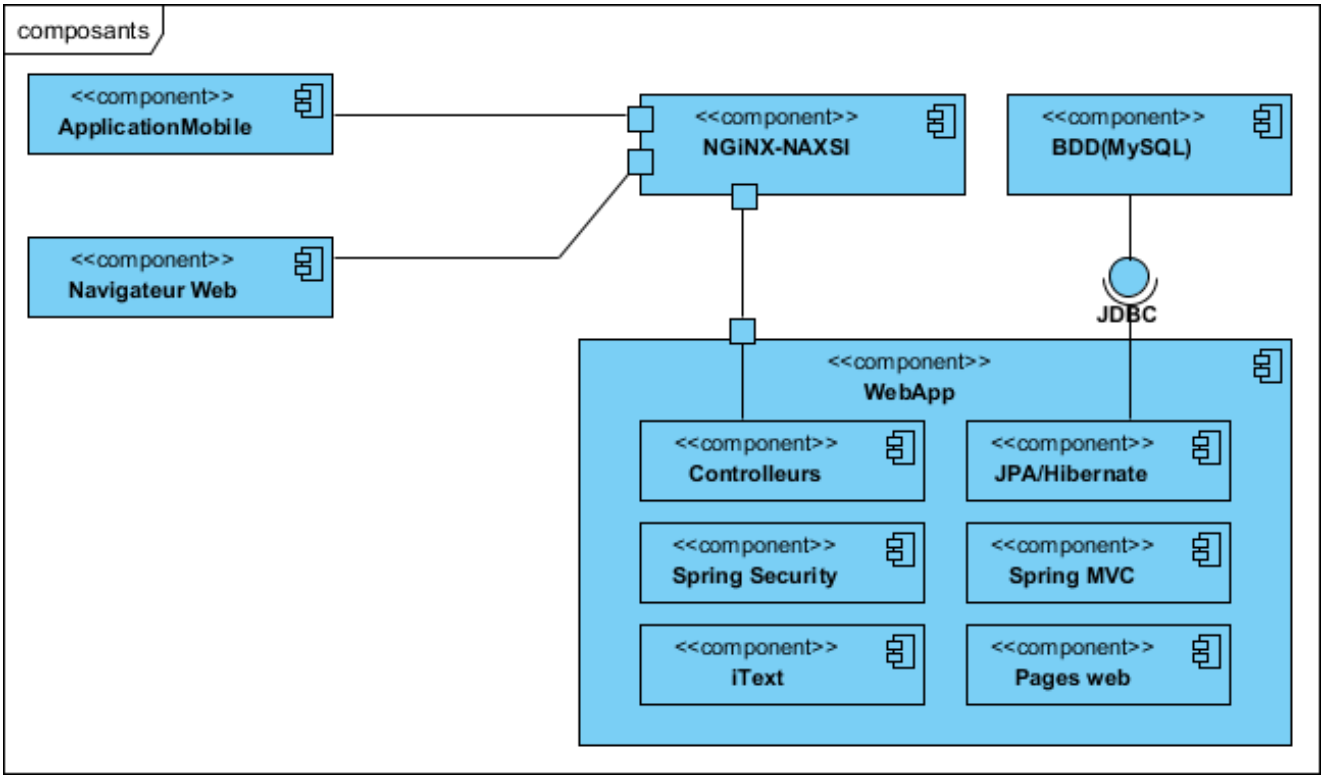


FIGURE 15: DIAGRAMME DES COMPOSANTS

II.6.3. Diagramme de déploiement

Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels. Les ressources matérielles sont représentées sous forme de nœuds qui, à leur tour sont connectés entre eux, à l'aide d'un support de communication. La nature des lignes de communication et leurs caractéristiques peuvent être précisées [7].

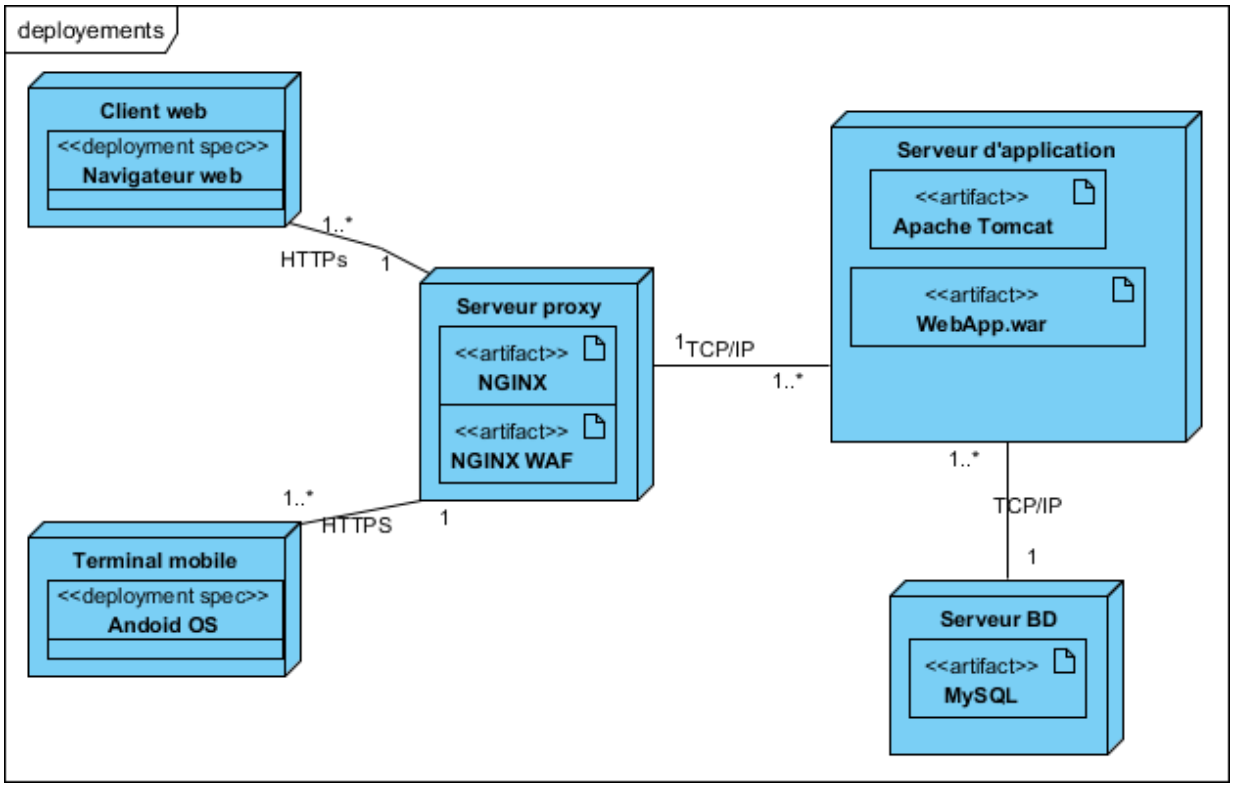


FIGURE 16: DIAGRAMME DE DEPLOIEMENT

Conclusion

Ce chapitre nous a permis de modéliser le futur système de gestion automatisée et d'optimisation du processus de recensement pré-électoral. Grâce au langage de modélisation UML, nous avons pu découvrir et représenter les différents aspects fonctionnels, comportementaux et architecturaux de notre système.

CHAPITRE 3 : MODELES ET METHODES D'OPTIMISATION DU PROCESSUS D'ENROLEMENT

Les modèles mathématiques permettent de modéliser, c'est-à-dire de représenter, toutes sortes de situations, d'objets et de structures du monde réel dans un but d'analyser et optimiser les paramètres d'un système [dans notre cas système de gestion et optimisation du processus de recensement] donné. Plusieurs outils mathématiques existent pour étudier le comportement des systèmes informatiques, nous pouvons citer l'algèbre relationnelle, théorie des graphes, réseau des neurones, files d'attente, réseaux de Pétri …

Dans ce chapitre, nous allons introduire les processus stochastiques ainsi que ses variantes entre autres les chaines de Markov, processus de naissance et de mort et les systèmes à files d'attente pour afin modéliser et étudier le comportement de notre système à long terme vis-à-vis des flux de candidats électeurs qui viendront se faire inscrire.

III.1. Processus stochastiques et chaines de Markov a temps continu

Les processus aléatoires ou stochastiques ont été conçus pour modéliser l'évolution temporelle de phénomènes et de systèmes aléatoires. Ils sont décrits par de familles discrètes ou continues de variables aléatoires $(X_t)_{t\in \mathrm{T}}$ où T est l'ensemble de temps d'observation des états du processus [30].

Un processus aléatoire est une famille de variables aléatoires $(X_t)_{t\in \mathrm{T}}$, indexée par l'ensemble T des temps dénombrable ou continu, définie sur un espace de probabilité $(\Omega, \mathcal{F}, \mathcal{P})$ et a valeurs dans un espace d'états E. La variable aléatoire X_t décrit l'état du processus au temps t. A toute occurrence ω on fait correspondre la trajectoire de la réalisation $(X_t(\omega))_{t\in T}$ du processus défini par l'application : $t \in T \longrightarrow X_t(\omega) \in E$ [30].

Le processus aléatoire $(X_t)_{t\geq 0}$ d'espace d'états $E = \{e_i\}_{i\in I}$ fini ou dénombrable, est une chaine de Markov a temps continu (c.m.c), si sont vérifiées les deux propriétés : [30]

(1) Propriété de Markov : $\forall (e_1, e_2, \dots, e_n, e_{n+1}) \in E^{n+1}, \forall (t_1, t_2, \dots, t_n, t_{n+1}) \in \mathbb{R}_+^{n+1}$ tels que $t_1 < t_2 < \cdots < t_n < t_{n+1}$

$$\boldsymbol{P\left(X_{t_{n+1}} = e_{n+1} \backslash X_{t_n} = e_n, \dots, X_{t_1} = e_1\right) = P\left(X_{t_{n+1}} = e_{n+1} \backslash X_{t_n} = e_n\right)} \quad \textbf{(1)}$$

(2) Homogénéité : $\forall t_1, t_2, t \in \mathbb{R}_+, \forall e_i, e_j \in E$:

$$\boldsymbol{P\left(X_{t_1+t} = e_j \backslash X_{t_1} = e_i\right) = P\left(X_{t_2+t} = e_j \backslash X_{t_2} = e_i\right) = p_{i,j}(t)} \quad \textbf{(2)}$$

La matrice de transition $P(t) = (p_{i,j}(t))_{i,j \in I}$ est une matrice stochastique et vérifie les deux propriétés :

(1) $\forall t, \forall i, j \in I, p_{i,j}(t) \geq 0$;

(2) $\forall t, \forall i \in I, \sum_j p_{i,j}(t) = 1$

III.1.1. Hypothèses de continuité et de dérivabilité des $p_{i,j}(t)$

Pour construire les concepts fondamentaux de générateur infinitésimal et d'équations de Kolmogorov associées à toute chaine de Markov a temps continu, on supposera vérifiées les hypothèses suivantes de continuité et de dérivabilité des $p_{i,j}(t)$ [30].

Hypothèse 1 : $\forall i \; et \; j, i \neq j \; \lim_{t \to 0^+} p_{i,j}(t) = 0$ et $\forall i, \lim_{t \to 0^+} p_{i,i}(t) = 1$

Hypothèse 2 : $\forall i \; et \; j, i \neq j \; \lim_{t \to 0} \frac{p_{i,j}(t) - p_{i,j}(0)}{t}$ existe et égale a $p'_{i,j}(0) = a_{i,j}$ dénommée taux de transition instantané de e_i a e_j ; $a_{i,j}$ est strictement positif si la transition de e_i a e_i est possible, car la probabilité de transition de e_i a e_i croit en fonction du temps.

Hypothèse 3 : $\forall i \; \lim_{t \to 0} \frac{p_{i,i}(t) - p_{i,i}(0)}{t}$ existe et égale a $p'_{i,i}(0) = a_{i,i}$ dénommée taux de permanence $a_{i,i}$ en l'état e_i est négatif, puisque contrairement à la situation précédente, la probabilité de séjourner dans un état quelconque décroît avec le temps.

Selon les hypothèses précédentes, la matrice $A = (a_{i,j})_{i,j}$ définie par $\lim_{t \downarrow 0} \frac{P(t) - 1_d}{t} = \frac{d}{dt}(P(t))_{(t=0^+)}$ existe ; elle est nommée générateur infinitésimal de la c.m.c $(X_t)_{t \geq 0}$

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots\cdots & a_{1,n} \\ a_{2,1} & \cdots & \cdots\cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots\cdots & a_{n,n} \end{pmatrix}$$

THEOREME : POUR TOUT ETAT $\boldsymbol{i, a_{i,i} = -\sum_{j+i} a_{i,j}}$ (3)

Théorème : Tout processus à accroissements indépendants vérifie la propriété de Markov.

Soit $(X_t)_{t\geq 0}$ une c.m.c telle que $X_0 = 0$; désignons X_{t_j} par X_j

$P(X_n < x_n \backslash X_1 = x_1, \dots, X_{n-1} = x_{n-1}) = P(X_n - X_{n-1} < x_n - x_{n-1} \backslash X_1 = x_1, \dots, X_{n-1} = x_{n-1}) = P(X_n - X_{n-1} < x_n - x_{n-1} \backslash X_1 - X_0 = x_1, \dots, X_{n-1} - X_{n-2} = x_{n-1} - x_{n-2}) = P(X_n - X_{n-1} < x_n - x_{n-1})$

car les $(X_j - X_{j-1})$ sont mutuellement indépendants.

$(X_n - X_{n-1})$ étant indépendant du vecteur$(X_1 - X_0, X_2 - X_1, \dots, X_n - X_{n-2})$, est donc indépendant de la somme de ses composantes. En conclusion, l'expression précédente est égale a

$$\boldsymbol{P(X_n - X_{n-1} < x_n - x_{n-1} \backslash X_{n-1} = x_{n-1}) = P(X_n < x_n \backslash X_{n-1} < x_{n-1})} \textbf{ (4)}$$

III.1.2. Equations de Kolmogorov

(1) Pour tout $t \geq 0$, la matrice de transition $P(t)$ d'une c.m.c vérifie les équations différentielles ci-dessous :

(a) Equations inverses ou du passé :

$\forall \boldsymbol{i,j}\ \boldsymbol{p'_{i,j}(t) = \sum_{k\in I} a_{i,k} p_{k,j}(t)}$, SOIT $\boldsymbol{P'(t) = A.P(t)}$ **(5)**

(b) Equations directes ou du futur :

$\forall \boldsymbol{i,j}\ \boldsymbol{p'_{i,j}(t) = \sum_{k\in I} p_{i,k}(t) a_{k,j}}$, SOIT $\boldsymbol{P'(t) = P(t).A}$ **(6)**

(2) Lorsque le nombre d'états est fini, $P(t)$ est définie par :

$$\boldsymbol{P(t) = \sum_{n\geq 0} \frac{(tA)^n}{n!} = e^{tA}} \textbf{ (7)}$$

III.1.3. Loi d'évolution et équation d'équilibre d'une chaine de Markov a temps continu

(1) Pour tout temps t le vecteur d'évolution $\pi(t)$ vérifie l'équation $\pi(t) = \pi(0)P(t)$ ou

$\forall \boldsymbol{i, \pi_i(t) = P(X_t = e_i)}$ **ET** $\boldsymbol{\pi(t) = (\pi_1(t), \dots, \pi_n(t))}$ **(8)**

(2) $\pi(t)$ est solution de l'équation d'évolution :

$\boldsymbol{\pi'(t) = \pi(t).A}$ **(9)**

Une distribution de probabilité $\pi = (\pi_i)_i$ portée par les états $(e_i)_i$ est stationnaire si pour tout j et pour tout $t \geq 0$: $\sum_i \pi_i\, p_{i,j}(t) = \pi_j$ qui s'écrit aussi $\pi.P(t) = \pi$

Soit une c.m.c de générateur A, si π est une distribution stationnaire, elle satisfait l'équation d'équilibre $\pi . A = 0$, équivalente aux équations :

$\forall j, \sum_{i \neq j} \pi_i a_{i,j} = -\pi_j a_{i,j}$, qui expriment l'égalité, en régime stationnaire, du flux moyen entrant dans l'état e_j et du flux moyen sortant vers l'ensemble des autres états. On la détermine en résolvant le système linéaire formé de l'équation d'équilibre et de l'équation de normalisation $\sum_i \pi_i = 1$

III.2. Processus de naissance et de mort

Un processus de naissance et de mort est une c.m.c a valeurs dans $\mathbb{N}$, dont le générateur infinitésimal $A = (a_{i,j})_{i,j}$ vérifie :

(1) $a_{i,j} = 0 \; si \; |i - j| \geq 2$;

(2) $a_{i,i+1} = \lambda_i$ est le taux de croissance ou de naissance de l'état e_i, λ_0 étant égal a $a_{0,1}$

(3) $a_{i,i-1} = \mu_i$ est le taux de décroissance ou de mort de l'état e_i, pour tout i non nul.

Le générateur A d'un processus de naissance et de mort $(X_t)_t$ est défini par la matrice diagonale :

$$\boldsymbol{A} = \begin{pmatrix} -\boldsymbol{\lambda_0} & \boldsymbol{\lambda_0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots \\ \boldsymbol{\mu_1} & -(\boldsymbol{\lambda_1} + \boldsymbol{\mu_1}) & \boldsymbol{\lambda_1} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \boldsymbol{\mu_2} & -(\boldsymbol{\lambda_2} + \boldsymbol{\mu_2}) & \boldsymbol{\lambda_2} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mu_3} & -(\boldsymbol{\lambda_3} + \boldsymbol{\mu_3}) & \boldsymbol{\lambda_3} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mu_4} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \textbf{(10)}$$

Diagramme de transition :

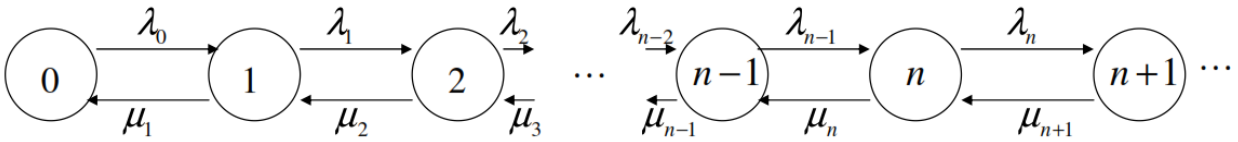


FIGURE 17: SCHEMA ILLUSTRATIF DU MODELE DU PROCESSUS DE NAISSANCE ET DE MORT

Théorème : Les probabilités $(\pi_n(t))_{n \in \mathbb{N}}$ vérifient le système d'équations différentielles

$$\begin{cases} \boldsymbol{\pi_0'(t) = -\lambda_0 \pi_0(t) + \mu_1 \pi_1(t);} \\ \boldsymbol{\forall n \geq 1, \pi_n'(t) = \lambda_{n-1} \pi_{n-1}(t) - (\lambda_n + \mu_n)\pi_n(t) + \mu_{n+1} \pi_{n+1}(t)} \end{cases} \quad \textbf{(11)}$$

La distribution stationnaire $\pi = (\pi_0, \pi_1, \dots, \pi_n, \dots)$ est solution de l'équation d'équilibre $\pi . A = 0$ équivalente au système linéaire :

$$\begin{cases} -\lambda_0 \pi_0 + \mu_1 \pi_1 = 0 \\ \forall n \geq 1, \lambda_{n-1} \pi_{n-1} - (\lambda_n + \mu_n)\pi_n + \mu_{n+1} \pi_{n+1} = 0 \end{cases}$$

La distribution stationnaire $(\pi_n)_n$ existe et est unique si et seulement si la série $\sum_{n \geq 1} \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n}$ est convergente. Dans ce cas, elle est définie par :

$$\boldsymbol{\pi_0 = (1 + \sum_{n \geq 1} \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n})^{-1}} \textbf{ ET } \boldsymbol{\forall n \geq 1, \pi_n = \pi_0 \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n}} \textbf{(12)}$$

III.3. Théories sur les files d'attentes et ses applications

Un système de service est constitué d'un espace d'attente ou les clients arrivent l'un après l'autre, en formant une file, et d'un espace de service [30]**.**

Il est décrit par les caractéristiques suivantes :

(1) La nature du processus stochastique décrivant le flux des entrées, indépendantes entre elles, déterminée par la loi des intervalles de temps entre deux arrivées. Si la loi est exponentielle, le flux est décrit par le processus de poisson d'intensité λ, qui est le taux moyen d'entrée des clients dans l'unique file d'attente.

(2) Le nombre s de services mis en parallèle

(3) La loi de probabilité de la durée de service S, de moyenne $\frac{1}{\mu}$; les durées de service sont toujours des variables aléatoires indépendantes de même loi, qui est souvent mais non nécessairement exponentielle.

(4) La capacité K du système, nombre maximal de clients présents dans le système

(5) La discipline d'attente ou politique de service : la règle la plus fréquemment utilisée s'exprime sous la forme le premier arrivé est le premier servi », connue sous l'abréviation anglo-saxonne FIFO (First In First Out).

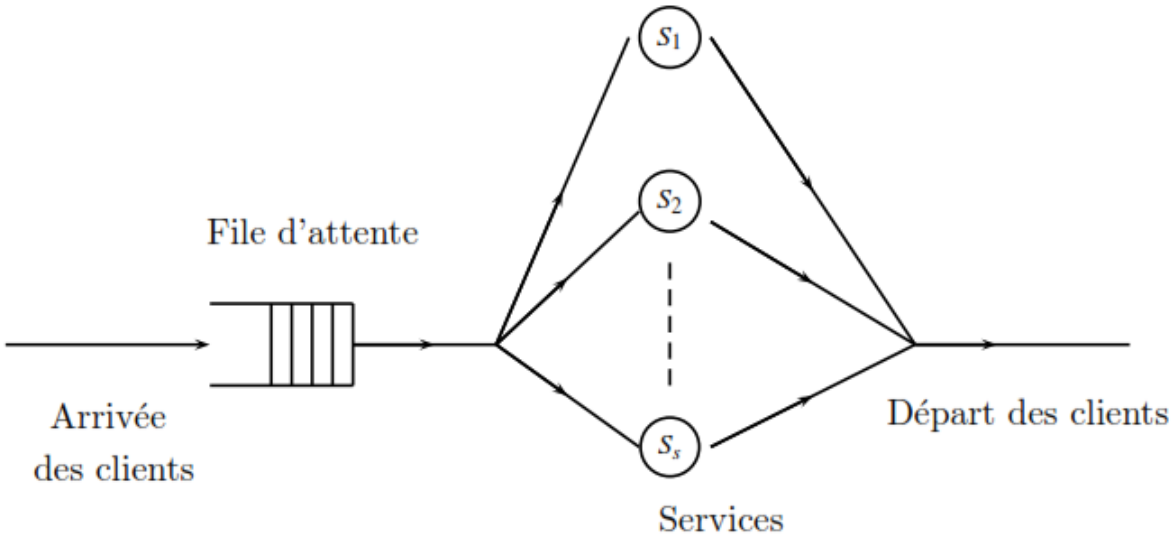


FIGURE 18: SCHEMA ILLUSTRATIF DU MODELE DE FILES D'ATTENTE

III.3.1. Notation de KENDALL Georges

Mise en place en 1953 par Georges David KENDALL, la notation de Kendall permet de symboliser une file d'attente par la suite de caractères A/S/s/K.

La lettre A caractérise la loi du processus d'arrivée (entrée) et S la loi de service. Par exemple, si on affecte la lettre M en A ou en S, on indique que le flux des entrées est poissonnier ou que les temps de service sont de loi exponentielle [30].

Ainsi, la lettre M a été choisie par Kendall pour exprimer le caractère markovien d'un processus dont les temps entre deux arrivées sont indépendants et de loi exponentielle. La lettre s dans la notation de Kendall désigne le nombre de service et la lettre K la capacité du système qui s'exprime par un entier strictement positif [30].

Ces quatre symboles peuvent être suivis par un cinquième qui qualifie la politique du service. On peut adjoindre à la nomenclature précédente deux symboles supplémentaires désignant le nombre maximal de clients potentiels et le sigle symbolisant la discipline de service [30].

III.3.2. Types de files d'attente

- *Files séparées* **:** une file par guichet (par exemple, aux caisses des supermarchés) ; ce système a l'inconvénient de générer des frustrations lorsque certaines files sont plus rapides que d'autres, ou lorsqu'un guichet supplémentaire s'ouvre, permettant aux derniers de passer les premiers [39] ;
- *File distribuée ou mutualisée* **:** une seule file alimente plusieurs guichets [39] ;
- *File virtuelle* **:** une prise de ticket permet de conserver l'ordre d'arrivée, sans avoir à faire la queue physiquement ; par exemple, les personnes peuvent s'asseoir en attendant leur tour [39] ;
- *File virtuelle mobile* **:** les nouvelles technologies permettent maintenant de prendre rang par internet ou par téléphone, et d'être prévenu par SMS lorsque son tour approche, le temps d'attente ne nécessitant plus une présence physique [39] ;
- *File prioritaire* **:** des files plus rapides peuvent être créées, par exemple pour les personnes ayant un handicap, ou pour les personnes ayant une carte de fidélité ; parfois, des files prioritaires payantes peuvent être proposées [39].

III.3.3. Etude du régime stationnaire des files d'attente

L'étude du régime transitoire des systèmes d'attente généraux est généralement très difficile, voire analytiquement impossible, sauf dans quelques cas comme le système M/M/1. Il s'agit d'un moindre mal du point de vue pratique, dans ce domaine comme beaucoup d'autres, seuls les régimes stationnaires importent [30].

Soit X_t la variable aléatoire associée au nombre de clients présents à l'instant t dans le système de service et $p_n(t)$ la probabilité $P(x_t = n)$. S'il existe un régime stationnaire, on le décrit par la variable aléatoire X indépendante de t, et on note sa distribution stationnaire par $(P(x = n) = p_n)_{n\in\mathbb{N}}$

Paramètres descriptifs d'un système d'attente

$\bar{\lambda}$ =Taux moyen d'entrée

$\frac{1}{\mu}$ = Temps moyen de service d'un client

L = Nombre moyen $E(x) = \sum_{n\geq 1} np_n$ de clients dans le système

L_q= Nombre moyen $\sum_{n\geq 1+s}(n-s)\,p_n$ de clients dans la file

W = Durée moyenne de présence d'un client quelconque dans le système

W_q = Durée d'attente moyenne d'un client quelconque dans la file

ρ = Intensité $\frac{\lambda}{\mu}$ du trafic dont l'unité est appelée Erlang

u = Taux d'utilisation de chaque serveur (proportion du temps pendant lequel il est occupé)

Les paramètres $L, L_q, W, W_q \; et \; \mu$ sont appelés paramètres de performance du système

III.3.4. Relation entre les paramètres de performance

$\boldsymbol{L = L_q + \frac{\bar{\lambda}}{\mu}}$ (13)

$\boldsymbol{L = \bar{\lambda} W}$(FORMULE DE LITTLE) (14)

$\boldsymbol{W = W_q + \frac{1}{\mu}}$ (15) $\boldsymbol{L_q = \bar{\lambda}.W_q}$(16)

III.4. Etude de la file M/M/1

Le système d'attente le plus simple est constitué d'un serveur et d'une file à capacité illimitée, désignée par M/M/1 ou M/M/1/∞. C'est un processus de naissance et de mort de paramètres $\lambda_n = \lambda$ si $n \neq 0$ et $\mu_0 = 0$. Dans ce cas, le taux moyen d'entrée $\bar{\lambda}$ est égal a λ

Diagramme de transition de la file M/M/1 :

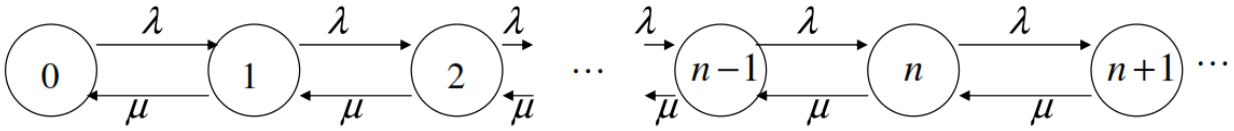


FIGURE 19: SCHEMA DU MODELE DE FILE D'ATTENTE M/M/1

III.4.1. Système de service M/M/1

(1) Le processus $(T_n)_{n\epsilon\mathbb{N}}$ des temps d'arrivée des clients est supposé poissonnien, d'intensité λ ; les durées $U_n = T_{n+1} - T_n$ entre deux arrivées consécutives, sont donc indépendants et de loi exponentielle $\mathcal{E}(\lambda)$

(2) Les durées de service $(S_n)_{n\epsilon\mathbb{N}}$ représentées par des variables aléatoires indépendantes sont de même loi exponentielle de paramètre μ définissant le taux de service ; la durée moyenne de service est donc égale a $\frac{1}{\mu}$.

On supposera vérifiée, la condition d'existence d'un régime stationnaire : l'intensité ρ du Trafic est strictement inferieur a 1.

La distribution stationnaire $(\pi_j)_j$ est de loi géométrique

$$\forall \boldsymbol{j}, \boldsymbol{\pi_j} = (\mathbf{1} - \boldsymbol{\rho})\boldsymbol{\rho^j} \textbf{ (17)}$$

Ainsi, $\forall n, \lambda_n = \lambda, \mu_n = \begin{cases} 0 \; si \; n = 0 \\ \mu \; si \; n \neq 0 \end{cases}$

D'autre part $\pi_n = \pi_0 \rho^n$ d'où $\pi_n = \rho^n(1-\rho)$

La durée d'attente T d'un client dans le système est de loi exponentielle de paramètre $(\mu - \lambda)$

III.4.2. Caractéristiques du système M/M/1

$$\boldsymbol{L = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda}} \textbf{ (18)} \quad \boldsymbol{L_q = \frac{\rho^2}{1-\rho} = \frac{\lambda^2}{\mu(\mu-\lambda)}} \textbf{ (19)}$$

$$\boldsymbol{W = \frac{1}{\mu-\lambda}} \textbf{ (20)} \qquad \boldsymbol{W_q = \frac{L}{\mu} = \frac{\lambda}{\mu(\mu-\lambda)}} \textbf{ (21)}$$

Par définition, $L = \sum_{n\geq 1} n\pi_n = \frac{\rho}{1-\rho}$ et $L_q = \sum_{n\geq 2}(n-1)\,\pi_n = L - \rho = \frac{\rho^2}{1-\rho}$

III.5. Etude des systèmes d'attente M/M/s/K

On rappelle que la capacité est le nombre maximal de clients en attente ou en service. Ce modèle convient aux situations où l'espace d'attente est de très grande taille, donc d'une capacité beaucoup plus grande que le nombre de serveur s.

III.5.1. Caractéristiques du système M/M/s/K

La condition d'existence d'un régime stationnaire :

$$\boldsymbol{\rho = \frac{\lambda}{\mu} < s} \textbf{ (22)}$$

$$\boldsymbol{\pi_0 = \left(\sum_{k=0}^{s-1} \frac{\rho^k}{k!} + \frac{\rho^s}{(s-1)!}\frac{1}{(s-\rho)}\right)^{-1}, \pi_k = \begin{cases} \frac{\rho^k}{k!}\pi_0 \; si \; k \leq s \\ \frac{\rho^k}{s!S^{k-s}}\pi_0 \; si \; k \geq s \end{cases}} \textbf{ (23)}$$

$$\boldsymbol{L_q = \frac{\pi_0 \rho^{s+1}}{(s-1)!(s-\rho)^2}} \textbf{ ; } \boldsymbol{W_q = \pi_0 \frac{\rho^s}{\mu(S-1)!(s-\rho)^2}} \textbf{ (24)}$$

La probabilité pour que tous les services soient occupés est égale a :

$$\pi_0 \frac{s\rho^s}{s!(s-\rho)} \textbf{ (25)}$$

Diagramme des transitions :

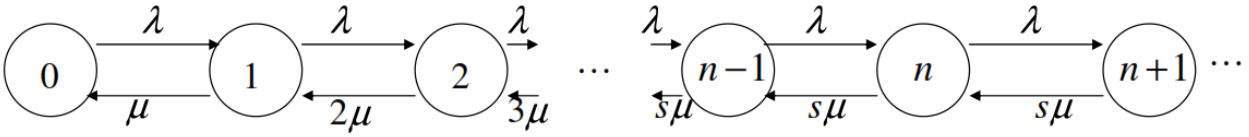


FIGURE 20: SCHEMA DU MODELE D FILE D'ATTENTE M/M/S/K

III.5.2. Etude du système M/M/s/K

Equations bilans :

$$\begin{cases} \lambda\pi_0 = \mu\pi_1 \\ (\lambda + k\mu)\pi_k = (k+1)\mu\pi_{k+1} + \lambda\pi_{k+1} \quad pour\ 0 < k < s \\ (\lambda + s\mu)\pi_k = s\mu\pi_{k+1} + \lambda\pi_{k-1} \ pour\ k \geq s \end{cases} \textbf{ (26)}$$

$$\begin{cases} \pi_k = \frac{\rho^k}{k!}\pi_0 \ \ pour\ k = 0, 1, \dots, s \\ \pi_k = \frac{\rho^k}{s!s^{k-1}}\pi_0\ pour\ k \geq s+1 \end{cases} \quad avec\ \rho = \frac{\lambda}{\mu} \text{ ET } \pi_0 = (\sum_{k=0}^{s-1} \frac{\rho^k}{k!} + \left(1 - \frac{\rho}{s}\right)^{-1} \frac{\rho^s}{s!})^{-1} \textbf{ (27)}$$

$$L_q = \sum_{k=s+1}^{+\infty}(k-s)\pi_k = \frac{s^s}{s!}\pi_0 \sum_{k\geq s+1}(k-s)\left(\frac{\rho}{s}\right)^k = \frac{s^s}{s!}\pi_0 \sum_{m=1}^{+\infty}\left(\frac{\rho}{s}\right)^{m+s} = \left(\frac{\rho}{s}\right)^{s+1}\frac{s^s}{s!}\left(1-\frac{\rho}{s}\right)^{-2} = \frac{\rho^{s+1}}{(s-1)!}\frac{\pi_0}{(s-\rho)^2} \text{ ET } W_q = \frac{L_q}{\lambda} \textbf{ (28)}$$

En prenant $s = 1$, on retrouve les caractéristiques du système M/M/1.

III.6. Système d'attente M/G/1

Ce système d'attente est caractérisé par une durée de service de loi quelconque symbolisée par G en notation de Kendall. On considère donc que le processus d'arrivée est toujours poissonnier d'intensité λ et que les temps de service T_S de densité f , d'espérance $\frac{1}{\mu}$ et de variance σ^2 sont indépendants entre eux. Il s'agit déterminer la distribution stationnaire du processus $(X_t)_t$.

Soit $(X_{T_i})_i$ la chaine incluse représentant le nombre de clients dans le système aux instants aléatoires respectifs $T_1, T_{2,}$ ..., ou le premier client, respectivement le second client quittent le service.

Soit N_k le nombre de clients entrant dans le système pendant la durée de service du $k - eme$ client ; les N_k sont indépendantes entre elles, de même loi :

$$\boldsymbol{P(N_k = n) = \int P(N_k = n \backslash T_S = t) f(t) dt = \int e^{-\lambda t} \frac{(\lambda t)^n}{n!} f(t) dt = a_n} \quad \textbf{(29)}$$

Les durées de service n'étant plus de loi exponentielle, le processus $(X_t)_t$ n'est plus markovien, alors que la chaine $(X_{T_i})_i$ l'est, de matrice de transition $P = (p_{i,j})_{i,j}$ définie par :

$$\begin{cases} \boldsymbol{p_{0,j} = a_j \; si \; j \geq 0} \\ \boldsymbol{p_{i,j} = a_{j-i+1} \; si \; 1 \leq i \leq j+1} \\ \boldsymbol{p_{i,j} = 0 \; sinon} \end{cases} \quad \textbf{(30)}$$

III.7. Etude du comportement de notre système à l'aide des files d'attente

Dans cette section nous allons étudier le comportement de notre système vis-à-vis des flux des candidats électeurs qui viendront se faire inscrire et passerons par la suite à la détermination des paramètres de performance entre autres le temps d'attente, le nombre de candidats électeurs dans le système, ….

Notre système est assimilé à un système de files d'attente avec des serveurs parallèles. La file d'attente est constituée par les candidats électeurs qui viennent s'inscrire et les serveurs sont les agents recenseurs qui offre un service(=inscription) aux candidats électeurs.

Les clients (=candidats électeurs) arrivent suivant un processus markovien de taux d'arrivée moyen λ qui est de 8 personnes par heure. Le temps de service (=inscription d'un candidat électeur) est de 5 minutes. Ainsi, chaque bureau d'inscription est constitué de 5 agents recenseurs qui accueillent et inscrivent les candidats électeurs.

Ainsi, deux modèles sont possibles pour notre système :

(1) Système avec autant de files d'attente que de serveurs parallèles (N files d'attente et N serveurs parallèles = Files séparées)

(2) Système avec une file d'attente et N serveurs parallèles (1 file d'attente et N serveurs parallèles = File distribuée ou mutualisée)

L'étude des deux modèles de notre système va nous permettre de déterminer les paramètres de performance pour chaque modèle et enfin choisir le modèle le plus optimal.

III.7.1. Modèle avec files séparées (N files d'attente et N serveurs)

Pour ce cas, nous avons une file d'attente devant chaque serveur. Le taux d'arrivée étant de λ dans le cas général, ici nous aurons un taux d'arrivée de $\frac{\lambda}{N}$ tout en gardant le même taux de service qui vaut μ.

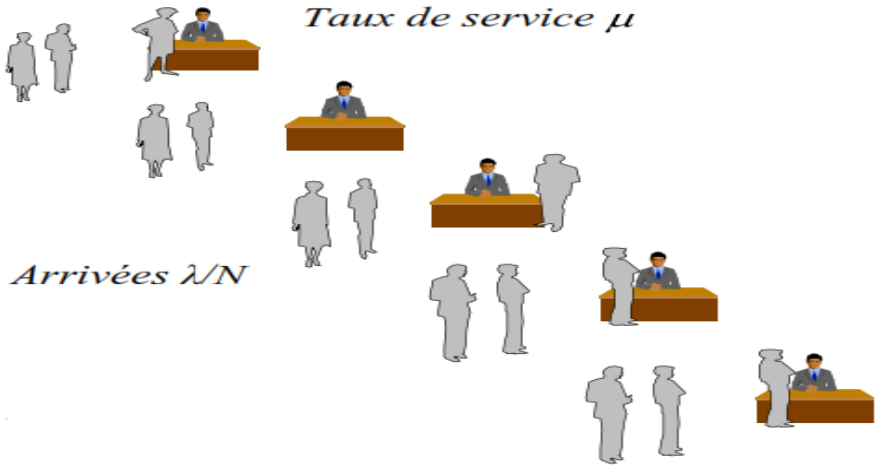


FIGURE 21: SCHEMA DU MODELE AVEC N FILES D'ATTENTE AVEC N SERVEURS PARALLELES

Le système étudié suivant ce modèle peut être décomposé en N sous-systèmes d'attente M/M/1. Ainsi, l'étude de ce modèle supposera que nous avons une série de systèmes M/M/1 qui opèrent comme un tout et par conséquent les théories des files d'attentes M/M/1 y seront appliquées.

Détermination des paramètres de performance de notre système

Nous avons déjà constaté que notre système peut être considéré comme un ensemble de N systèmes M/M/1. Donc, tous les calculs se feront comme nous avions un serveur unique.

Ainsi $\lambda' = \frac{\lambda}{N} = \frac{8}{5}$ est le taux moyen d'arrivée des candidats électeurs dans chaque sous-système, et $\mu = \frac{60}{5} = 12$ le taux de service pour chaque sous-système.

$$\rho = \frac{\lambda'}{\mu} = \frac{\frac{8}{5}}{12} = \frac{2}{15}\ erlang$$

Remarquons que l'intensité du trafic est strictement inférieure à 1, donc ce modèle est stationnaire.

$$L = \frac{\rho}{1-\rho} = \frac{\frac{2}{15}}{1-\frac{2}{15}} = \frac{2}{13} = 0{,}1538461539$$

$$L_q = \frac{\rho^2}{1-\rho} = \frac{\left(\frac{2}{15}\right)^2}{1-\frac{2}{15}} = \frac{4}{195} = 0{,}0205128205$$

$$W = \frac{1}{\mu-\lambda} = \frac{1}{12-\frac{8}{5}} = \frac{1}{260} = 0{,}0038461539$$

$$W_q = \frac{L}{\mu} = \frac{\frac{2}{13}}{12} = \frac{1}{78} = 0{,}0128205128$$

III.7.2. Modèle avec une file distribuée ou mutualisée

Le modèle avec une file distribuée ou mutualisée est, pour notre cas, un modèle avec une (1) seule file d'attente et un certain nombre (N) de serveurs parallèles. L'unique file d'attente que nous avons est desservie par les N serveurs parallèles. Le taux d'arrivée vaut λ et le taux de service vaut $N\mu$.

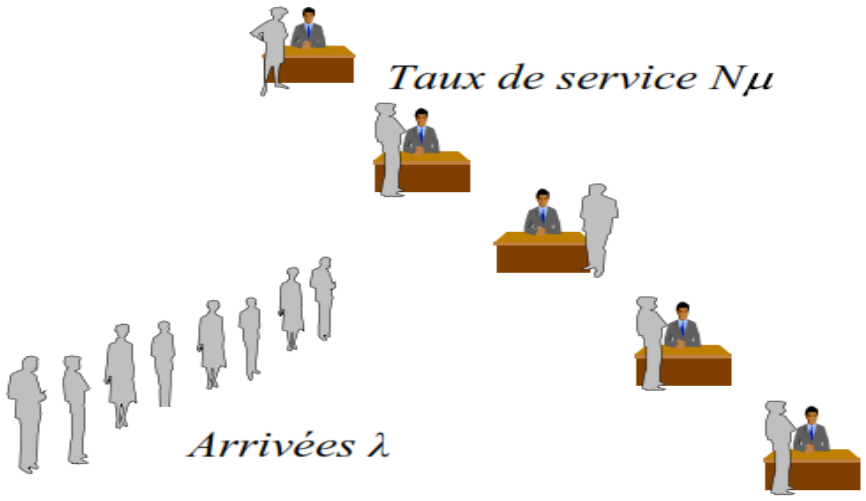


FIGURE 22: SCHEMA DU MODELE AVEC 1 FILE D'ATTENTE AVEC N SERVEURS PARALLELES

Ce modèle reflète le système de file d'attente M/M/N/K, dans lequel le processus des arrivées et celui des services sont markoviens avec N serveurs, et une capacité du système K. Donc, nous appliquerons les théories des systèmes M/M/N/K avec K tendant vers l'infini.

Détermination des paramètres de performance de notre système

La détermination des paramètres de notre système consiste à calculer
L : Le nombre moyen de candidats électeurs dans le système d'attente
L_q : Le nombre moyen de candidats électeurs dans la file d'attente
W : La durée moyenne de présence d'un candidat électeur quelconque dans le système
W_q: La durée moyenne de présence d'un candidat électeur quelconque dans la file
Le nombre de serveur $s = 5$ et $\mu' = s\mu = 5 * 12 = 60$, l'intensité du Trafic vaut :

$$\rho = \frac{\lambda}{\mu'} = \frac{8}{60} = \frac{2}{15}\ erlang$$

$$\pi_0 = \left(1 + \frac{2}{15} + \frac{\left(\frac{2}{15}\right)^2}{2!} + \frac{\left(\frac{2}{15}\right)^3}{3!} + \frac{\left(\frac{2}{15}\right)^4}{4!} + \left(1 - \frac{\frac{2}{15}}{5}\right)^{-1} \frac{\left(\frac{2}{15}\right)^5}{5!}\right)^{-1}$$

$$= 0{,}8753246384$$

$$L_q = \frac{0{,}8753246384 * \left(\frac{2}{15}\right)^6}{4!\left(5 - \frac{2}{15}\right)^2} = 0{,}0000006316$$

$$W_q = \frac{0{,}0000006316}{8} = 0{,}000000079$$

$$W = W_q + \frac{1}{\mu} = 0{,}000000079 + \frac{1}{60} = 0{,}0166667456$$

$$L = \lambda W = 8 * 0{,}0833334123 = 0{,}1333339648$$

III.7.3. Comparaison des paramètres de performance des deux modèles

L'étude des deux modèles de files d'attente à savoir : modèle à files mutualisées et le modèle à files séparées, nous a permis d'analyser les paramètres de performances de ces dernières.

TABLE 17: COMPARAISON DES PARAMETRES DE PERFORMANCE

Paramètres	Files séparées	Files mutualisées	Commentaire
ρ	$\frac{2}{15}$	$\frac{2}{15}$	L'intensité du trafic est la même pour tous les modèles
L_q	0,0205128205	0,0000006316	Le nombre moyen de clients sur la file est plus petit pour les files mutualisées
W_q	0,0128205128	0,000000079	De même, la durée moyenne d'attente d'un client sur la file est plus petite pour les files mutualisées
W	0,0038461539	0,0166667456	A durée d'attente dans le système est plus petite pour les files séparées
L	0,1538461539	0,1333339648	Le nombre moyen de client dans le système est petit pour les files mutualisées

La comparaison des paramètres de performance dans le tableau précédent montre que les files d'attentes à files mutualisées sont les mieux indiquées pour notre système étudié.

Conclusion

Dans ce chapitre, nous avons introduit les processus stochastiques comme outils mathématique de modélisation et étude de l'évolution des systèmes à temps discrets et continus. Et par la suite, avons passé à l'étude du comportement à long terme de notre système en comparant deux modèles possibles pour notre système.

L'étude du comportement de notre système à l'aide des files d'attentes nous a permis d'opérer un choix entre les deux modèles possibles de notre système (modèle de N files avec N serveurs et 1 file avec N serveurs). Bien que les deux modèles soient tous stationnaires, le tableau d'analyse des paramètres de performance des deux modèles nous prouvée que le deuxième modèle (à files mutualisées) est optimal.

CHAPITRE 4 : ANALYSE DE LA SECURITE DU NOUVEAU SYSTEME AUTOMATISE DE GESTION ET OPTIMISATION DU PROCESSUS DE RECENSEMENT

La sécurité des systèmes d'information représente l'ensemble de principes et bonnes pratiques qui permettent de garantir un accès sécurisé aux ressources matérielles et logicielles d'un système donné en tenant compte des droits et privilèges des utilisateurs [16].

Selon le W3C, une application Web est une application basée sur le protocole HTTP, indépendante des plates-formes et langages d'implémentation et reposant sur une architecture Web. Les technologies web reposent sur un ensemble de protocoles, langages et architectures ouverts dont les spécifications et standards sont librement accessibles [12].

Ainsi, la sécurité des applications web doit être prise en compte dès la conception, le développement, lors de l'intégration et durant toute la vie de l'application. Elle doit faire partie intégrante du cycle de vie du système. Au-delà de la conception et de l'implémentation, la sécurité d'une application web doit être testée et vérifiée avant la mise en production et lors de ses évolutions [16].

Dans ce chapitre nous allons explorer les failles et vulnérabilités du Top 10 de l'OWASP et leurs contre-mesures respectives, et par la suite présenter les différentes technologies utilisées pour mettre en place un système sécurisé. Open Web Application Security Project (OWASP) est une communauté publique permettant à des organismes de développer, acheter et maintenir des applications fiables [15].

IV.1. OWASP Top 10 2013 Risques de Sécurité des Applications web

1. Injection [15]

Une faille d'injection, telle l'injection SQL, OS et LDAP, se produit quand une donnée non fiable est envoyée à un interpréteur en tant qu'élément d'une commande ou d'une requête. Les données hostiles de l'attaquant peuvent duper l'interpréteur afin de l'amener à exécuter des commandes fortuites ou accéder à des données non autorisées.

Bien qu'une application web puisse subir différents types d'attaques par injection, il suffit de vérifier que les caractères utilisés sont ceux attendus. Ce contrôle doit être effectué au niveau du client grâce à JavaScript et au niveau du serveur lorsque les paramètres sont récupérés pour fermer la faille de sécurité.

2. Violation de Gestion d'Authentification et de Session [15]

Les fonctions applicatives relatives à l'authentification et la gestion de session ne sont souvent pas mises en œuvre correctement, permettant aux attaquants de compromettre les mots de passe, clés, jetons de session, ou d'exploiter d'autres failles d'implémentation pour s'approprier les identités d'autres utilisateurs.

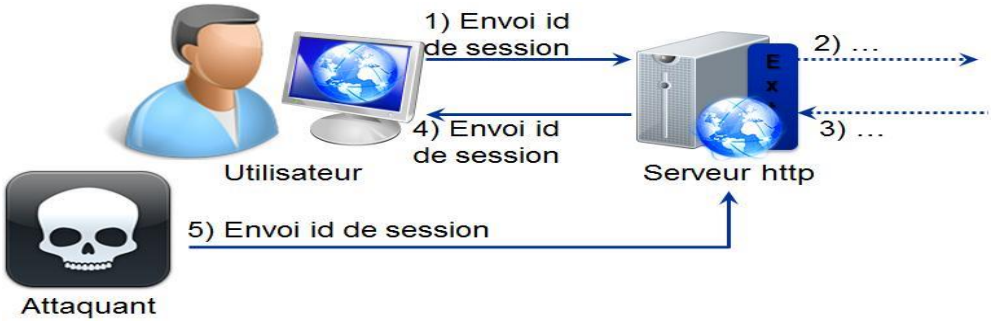


FIGURE 23: ILLUSTRATION DE L'ATTAQUE PAR VIOLATION GESTION D'AUTHENTIFICATION ET SESSION

Concernant les systèmes d'authentification, l'application ne doit accepter que des mots de passe suffisamment forts pour éviter d'être devinés rapidement par la force brute. Une longueur minimale de huit caractères est recommandée par l'OWASP pour les applications critiques. De plus il doit comporter au moins un chiffre, une lettre en minuscule et une lettre en majuscule.

Le message affiché lors d'un problème de validité de l'identifiant ou du mot de passe doit être générique et ne doit pas donner d'indice quant à l'origine de l'erreur. Dans la mesure du possible il est conseillé de ne pas développer son propre mécanisme d'authentification. Il est préférable d'utiliser un système existant éprouvé.

Du côté client, du code JavaScript doit fermer la session lorsque l'utilisateur ferme le navigateur. Cela permet de simuler une action de l'utilisateur pour se déconnecter de l'application.

Pour que l'utilisateur évite de perdre des saisies non sauvegardées, du code JavaScript peut prévenir l'utilisateur que sa session va bientôt expirer.

3. Cross-Site Scripting(XSS) [15]

Les failles XSS se produisent chaque fois qu'une application accepte des données non fiables et les envoie à un browser web sans validation appropriée. XSS permet à des attaquants d'exécuter du script dans le navigateur de la victime afin de détourner des sessions utilisateur, défigurer des sites web, ou rediriger l'utilisateur vers des sites malveillants.

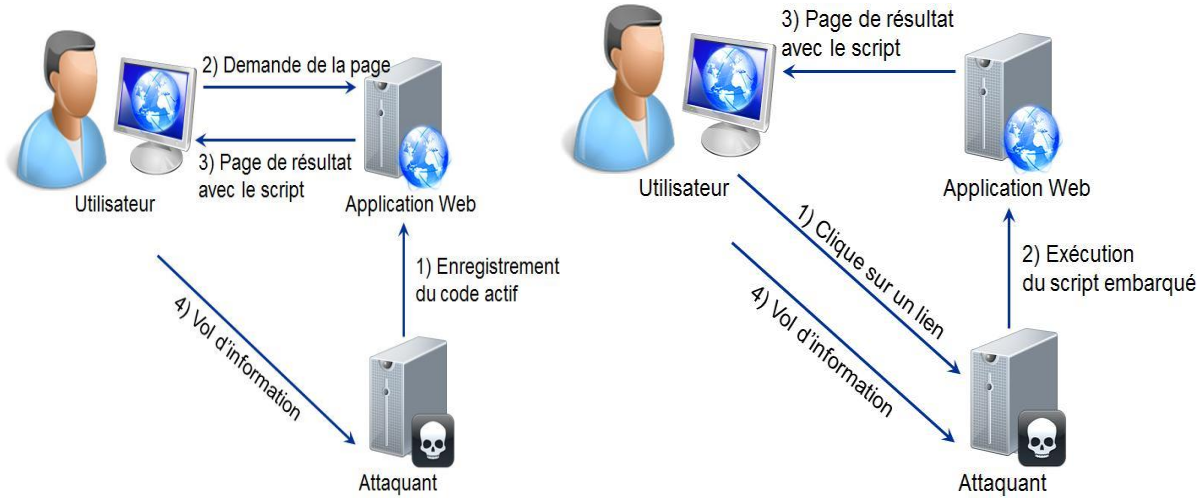


FIGURE 24 : ILLUSTRATION DE L'ATTAQUE PAR EXPLOITATION DES FAILLES XSS

Du côté client avec JavaScript il faut vérifier les données saisies par les utilisateurs. Côté serveur, il faut vérifier les données récupérées en paramètre. Il faut rejeter toutes les données qui ne sont pas conformes à ce qui est attendu.

Pour éviter le vol de cookies par du code JavaScript, il est possible de positionner l'attribut de cookie HTTPOnly. S'il est présent, le navigateur interdit au moteur JavaScript de lire ou écrire dans les cookies. Cet attribut est très peu utilisé par les applications Web, car tous les navigateurs ne le gèrent pas. Cependant il est préférable de l'utiliser car les navigateurs les plus populaires l'implémentent, ce qui diminue les risques liés aux cookies.

4. Références directes non sécurisées à un objet [15]

Une référence directe à un objet se produit quand un développeur expose une référence à un objet d'exécution interne, tel un fichier, un dossier, un enregistrement de base de données ou une clé de base de données. Sans un contrôle d'accès ou autre protection, les attaquants peuvent manipuler ces références pour accéder à des données non autorisées.

Pour protéger les données les plus confidentielles ou les fonctionnalités les plus avancées, le WASC recommande de demander à l'utilisateur de saisir à nouveau son identifiant et son mot avant de pouvoir y accéder. Ensuite il suffit de se baser sur ces valeurs pour construire les requêtes.

5. Mauvaise configuration Sécurité [15]

Une bonne sécurité nécessite de disposer d'une configuration sécurisée définie et déployée pour l'application, contextes, serveur d'application, serveur web, serveur de base de données et la plate-forme. Tous ces paramètres doivent être définis, mis en œuvre et maintenus, car beaucoup ne sont pas livrés sécurisés par défaut. Cela implique de tenir tous les logiciels à jour.

L'ANSSI et l'OWASP font les recommandations suivantes.

- Désactiver les options inutiles des composants afin de diminuer le nombre de vulnérabilités potentielles que ce soit au niveau du système d'exploitation, du système de gestion de bases de données ou du serveur HTTP.
- Mettre à jour les différents composants de l'architecture autant que possible en installant les correctifs dès qu'ils sont publiés. De plus, à l'installation il est préférable de choisir la version anglaise plutôt qu'une autre langue. En effet, lors du développement de correctifs c'est toujours la version anglaise qui est privilégiée, les autres versions étant corrigées plus tardivement.
- Désactiver voire supprimer tous les comptes inutiles. Le mot de passe des autres comptes doit être modifié dès que possible. Les comptes d'administration par défaut doivent être verrouillés. Il faut préférer l'utilisation de comptes d'administration créés manuellement.

6. Exposition de données sensibles [15]

Beaucoup d'applications web ne protègent pas correctement les données sensibles telles que les cartes de crédit, identifiants d'impôt et informations d'authentification. Les pirates peuvent voler ou modifier ces données faiblement protégées pour effectuer un vol d'identité, de la fraude à la carte de crédit ou autres crimes. Les données sensibles méritent une protection supplémentaire tel un chiffrement statique ou en transit, ainsi que des précautions particulières lors de l'échange avec le navigateur.

Pour nous prémunir de cette attaque, l'OWASP nous donne les recommandations suivantes :

- Identifier les menaces contre lesquelles l'on souhaite se protéger (ex. : menace interne, utilisateurs externes) et s'assurer que les données sensibles sont chiffrées correctement lors de leur stockage et de leur transport.
- Ne conserver que les données sensibles nécessaires. Les données que l'on ne possède pas ne peuvent être volées !
- Choisir des algorithmes éprouvés et générer des clés robustes. S'assurer qu'une gestion des clés est en place. Privilégier des modules cryptographiques certifiés.
- Stocker les mots de passe au moyen d'un algorithme adapté à cet usage, tel que bcrypt, PBKDF2, or scrypt.
- Désactiver la mise en cache et l'attribut "autocomplete" dans les formulaires collectant des données sensibles.

7. Manque de contrôle d'accès au niveau fonctionnel [15]

Pratiquement toutes les applications web vérifient les droits d'accès au niveau fonctionnel avant de rendre cette fonctionnalité visible dans l'interface utilisateur. Cependant, les applications doivent effectuer les mêmes vérifications de contrôle d'accès sur le serveur lors de l'accès à chaque fonction. Si les demandes ne sont pas vérifiées, les attaquants seront en mesure de forger des demandes afin d'accéder à une fonctionnalité non autorisée.

Votre application devrait utiliser un module de gestion des autorisations consistant, facilement analysable et appelé depuis les fonctionnalités métier. Ce type de protection est fréquemment fourni par des composants externes.

8. Falsification de requête inter-sites (CSRF) [15]

Une attaque CSRF (Cross Site Request Forgery) force le navigateur d'une victime authentifiée à envoyer une requête HTTP forgée, comprenant le cookie de session de la victime ainsi que toute autre information automatiquement inclue, à une application web vulnérable. Ceci permet à l'attaquant de forcer le navigateur de la victime à générer des requêtes dont l'application vulnérable pense qu'elles émanent légitimement de la victime.

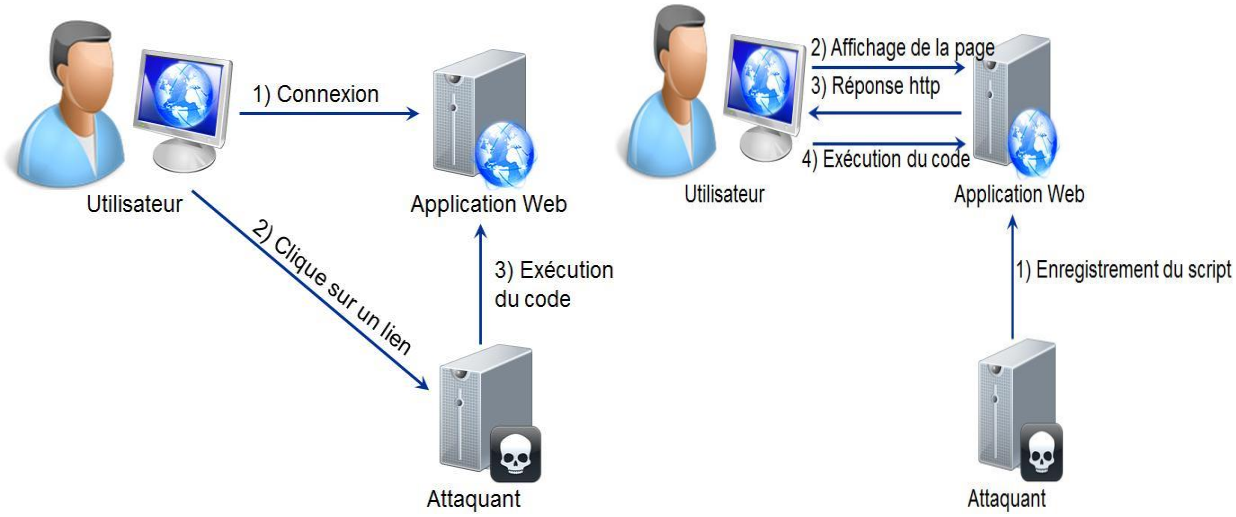


FIGURE 25: ILLUSTRATION DE L'ATTAQUE PAR CSRF

Bien que XSS et CSRF soient proches dans le principe, se protéger des attaques XSS ne permet pas de se protéger des attaques CSRF. Pour se protéger, il faut utiliser uniquement des requêtes POST. Les méthodes GET doivent être bannies. Attention toutefois, dans les servlet Java la méthode « doGet() » fait appel à la méthode « doPost() » en redirigeant l'ensemble des paramètres. Dans ce cas l'utilisation de requêtes GET fonctionne. C'est pourquoi l'utilisation de POST n'est pas une protection suffisante.

Pour les pages qui manipulent des données sensibles, il faut demander à l'utilisateur de s'authentifier à nouveau. Cela permet de s'assurer que l'utilisateur est conscient de l'action et l'approuve. L'utilisateur doit toujours vérifier que le lien sur lequel il clique est bien celui de l'application qu'il veut utiliser.

9. Utilisation de composants avec des vulnérabilités connues [15]

Les composants vulnérables, tels que bibliothèques, contextes et autres modules logiciels fonctionnent presque toujours avec des privilèges maximums. Ainsi, si exploités, ils peuvent causer des pertes de données sérieuses ou une prise de contrôle du serveur. Les applications utilisant ces composants vulnérables peuvent compromettre leurs défenses et permettre une série d'attaques et d'impacts potentiels.

Une option est de ne pas utiliser des composants que vous n'avez pas vous même écrit. Mais cela n'est pas très réaliste. De nombreux projets de composants ne fournissent pas de correctifs de sécurité pour les anciennes versions. A la place, le problème étant simplement corrigé dans la version suivante. De ce fait, effectuer une montée de version est critique.

Les projets logiciels devraient avoir un processus en place pour :

- Identifier tous les composants et les versions utilisées, ainsi que les dépendances (ex : le plugin des versions).
- Surveiller les bases de données publiques, les listes de diffusion des projets et les listes de diffusion de sécurité afin de s'assurer de la sécurité de ces composants afin de les maintenir à jour.
- Établir des politiques de sécurité pour l'utilisation des composants, telles que la mise en place de pratiques de développement sécurisé, le passage avec succès des recettes sécurité et l'utilisation de composants sous License.
- Ajouter des filtres de sécurités autour des composants afin de désactiver des fonctionnalités et/ou des parties comprenant des faiblesses ou des fonctions vulnérables.

10. Redirections et renvois non validés [15]

Les applications web réorientent et redirigent fréquemment les utilisateurs vers d'autres pages et sites internet, et utilisent des données non fiables pour déterminer les pages de destination. Sans validation appropriée, les attaquants peuvent réorienter les victimes vers des sites de phishing ou de malware, ou utiliser les renvois pour accéder à des pages non autorisées.

La redirection ne doit renvoyer qu'à des pages locales. Dans ce cas les caractères spéciaux doivent être prohibés. En cas de changement d'adresse ou de déplacement d'une page, il vaut mieux utiliser la redirection paramétrée dans le serveur HTTP. Par exemple avec Apache il est possible de placer un fichier « .htaccess » pour paramétrer les redirections automatiques.

IV.2. Pare-feu applicatif web : NAXSI pour NGINX

Un Web Application Firewall (pare-feu applicatif web), est une brique logicielle, généralement placée en amont d'un applicatif web, et qui a pour but d'analyser les requêtes HTTP arrivant à ce dernier. L'objectif est de détecter et prévenir l'exploitation d'éventuelles vulnérabilités présentes dans ledit applicatif, telles que celles décrites par le top 10 de l'OWASP [15]. Les pares-feux applicatifs sont aussi parfois utilisés pour réaliser du « virtual patching ». À l'inverse d'un filtrage global, il s'agit ici de protéger l'applicatif uniquement contre certaines vulnérabilités connues, en appliquant une liste blanche sur la partie spécifique du code posant problème.

La très grande majorité des WAF fonctionnent à la manière des antivirus, en s'appuyant sur une base de signatures, censés représenter les différents motifs d'attaques possibles. Une approche par liste blanche, moins répandue, est aussi possible. Cette approche remonte parfois plus de faux positifs, principale raison de son faible taux d'adoption. Mais en contrepartie, elle bénéficie de performances accrues et d'une sécurité renforcée car non dépendante d'une base de signatures [18].

Dans notre application, nous avons utilisé NAXSI pour NGINX, un module jouant le rôle de pare-feu applicatif intégré dans NGINX.

La majorité des pares-feux appliquant un « modèle négatif » (exclusion des mauvaises requêtes) reposent sur une base d'expressions régulières généralement appelées signatures. Ces expressions régulières (ou regex) sont ensuite recherchées dans le contenu de la requête HTTP afin de détecter une éventuelle attaque et de bloquer la requête.

Le reproche récurrent de ce type de fonctionnement est que ces expressions régulières sont souvent complexes et très nombreuses (souvent plusieurs centaines), ce qui peut entraîner un ralentissement de l'application. De plus, elles ont besoin d'être mises à jour régulièrement pour être efficaces, à la manière d'un antivirus.

Un autre problème, conceptuel, de cette approche est qu'il est nécessaire de « limiter » le champ de recherche de ces fameuses expressions régulières, ou d'appliquer des transformations avant de les effectuer. Dans le cas contraire, les performances seraient encore plus dégradées. Ces modifications entraînent l'apparition de risques de contournement liés aux attaques obfusquées.

L'objectif de NAXSI étant de s'affranchir de ces limites (performances & pérennité de la sécurité), il a fallu abandonner le système de signature « classique », pour s'approcher d'un mécanisme par liste blanche. NAXSI ne recherche donc pas de signatures complexes d'attaques, mais se concentre sur des primitives peu nombreuses (moins de 40), souvent réduites à des motifs d'un caractère ou deux, nécessaires à la majorité des attaques.

Cette approche permet notamment de limiter le temps de traitement nécessaire, ainsi que de le garder – dans une certaine mesure – prévisible. Le prix de ce mode opératoire qui limite le nombre de signatures, plus rapide et plus efficace contre les risques d'attaques offusquées/complexes, est une utilisation plus importante des listes blanches (ou exception, selon la terminologie). Pour pallier ce souci, NAXSI dispose d'un mode d'apprentissage, qui est au centre de son fonctionnement.

Un autre bénéfice de NAXSI est fourni par le fait qu'il repose sur Nginx. Grâce à cette architecture, NAXSI peut être utilisé aussi bien en tant que reverse proxy, qu'en tant que module sur le serveur web lui-même, sans pour autant impacter de manière significative les performances du serveur HTTP (moins d'1% de ressources).

L'architecture logicielle interne de NAXSI, est faite de plusieurs composants indépendants et remplaçables suivants :

2.1. Le core

Ce module Nginx (développé en C) est le seul à « agir » en runtime. Il inspecte les requêtes HTTP en fonction des règles spécifiées dans la configuration Nginx, généralement /etc/nginx/naxsi_core.rules3.

Le module core va lire (parser) les requêtes HTTP, comparer leur contenu avec les signatures NAXSI, puis, en fonction de la configuration faite par l'utilisateur et du /des scores de la requête, éventuellement rediriger la requête vers une location tierce afin de protéger l'applicatif.

2.2. Le démon d'interception « nx_intercept »

Le démon d'interception (script minimaliste en python, basé sur twisted) est en charge de stocker les exceptions. Une exception correspond au déclenchement d'une ou plusieurs règles.

Cette acquisition peut être faite depuis les fichiers de logs de Nginx (puisque NAXSI enregistre aussi les signatures d'attaques dans les error_log de Nginx), ou depuis les requêtes dupliquées reçues lorsque NAXSI est en mode apprentissage. Les données sont ensuite stockées dans une base de données (mysql ou sqlite3).

2.3. Le démon d'extraction nx_extract

Ce démon est lui aussi en python (et lui aussi basé sur twisted). En s'appuyant sur la base de données alimentée par le démon d'interception (à partir des exceptions interceptées), il est en charge de générer les listes blanches. Il est aussi utilisé pour générer des statistiques sur les interceptions réalisées. À la différence de nx_intercept, il a pour vocation d'être utilisé (lancé) par l'utilisateur.

IV.3. SSL et certificats SSL

SSL est un protocole de communication sécurisé intégré aux principaux navigateurs et serveurs Web, il utilise des techniques de cryptage qui s'appuient sur un système de clé publique/privée initialement développé sur RSA. L'établissement d'une connexion SSL nécessite l'installation d'un certificat numérique sur le serveur Web. Ce certificat utilise alors les clés publiques et privées pour le cryptage, et identifie le serveur de manière unique et définitive.

Les certificats numériques s'apparentent à une forme de carte d'identité électronique qui permet au client d'authentifier le serveur avant l'établissement d'un canal de communication crypté.

Un certificat SSL remplit quatre fonctions bien distinctes, toutes indispensables à la protection de la confidentialité et de la sécurité des clients et utilisateurs : cryptage, intégrité, authentification et non-répudiation.

Le cryptage utilise des algorithmes mathématiques pour transformer les données et les rendre exclusivement lisibles par les parties concernées. Dans le cadre d'une transaction sécurisée par SSL, les clés privées et publiques fournies avec le certificat numérique du serveur jouent un rôle déterminant dans la sécurisation des données envoyées et reçues par le navigateur Web.

En cryptant les données pour les rendre exclusivement lisibles par les parties concernées, les certificats SSL assurent également leur intégrité. En d'autres termes, ces données ne pouvant être lues par aucun tiers, leur modification en cours de transit s'avère par conséquent impossible. Si les données cryptées étaient modifiées, elles seraient de fait rendues inutilisables – ce qui ne saurait échapper aux parties concernées. La moindre tentative d'interférence est donc automatiquement repérée.

L'émission d'un certificat numérique par une autorité de certification permet essentiellement de valider l'identité de l'organisme – ou de la personne – à l'origine de la demande de certificat. Les certificats SSL sont associés à un nom de domaine Internet.

La vérification par l'AC de l'identité du propriétaire du domaine en question permet aux utilisateurs de savoir dans un premier temps à qui ils ont affaire. Ainsi, lorsque vous vous connectez sur un site sécurisé par SSL comme Amazon.fr, le certificat identifie le propriétaire comme étant la société Amazon, Inc. Vous avez ainsi l'assurance d'être sur le véritable site exploité par Amazon.

La non-répudiation se caractérise par l'association de fonctions de cryptage, de protection de l'intégrité et d'authentification. En clair, aucune partie prenante à une transaction sécurisée ne peut légitimement affirmer que ses échanges/communications provenaient d'une autre personne ou entité.

Cette caractéristique supprime toute possibilité pour l'une des parties de répudier ou de « se rétracter » par rapport à des informations communiquées en ligne.

Le protocole SSL utilise des clés de cryptage publiques et privées. Lors de l'émission d'un certificat numérique pour un serveur Web, ce certificat contient deux clés : l'une, détenue de manière privée par le serveur Web (« clé privée »), et l'autre, rendue publique à toute personne en faisant la demande (« clé publique »). Ces deux clés sont asymétriques :

- Les données cryptées par la clé privée ne peuvent être décryptées que par la clé publique
- Les données cryptées par la clé publique ne peuvent être décryptées que par la clé privée

Ainsi, pour garantir la confidentialité des échanges, le navigateur Web s'adresse à la clé publique du serveur. Le navigateur l'utilise ensuite pour crypter les informations à transmettre – informations qui pourront ensuite être décryptées par le serveur Web à l'aide de sa propre clé privée.

Dans la pratique, le processus de cryptage fait parfois intervenir des clés de session aléatoires, valables pour une courte durée entre le navigateur et le serveur. Ces clés de session sont utilisées car bien souvent, le navigateur ne possède pas son propre certificat numérique et sa propre paire de clés.

IV.4. Présentation de Spring Security

Spring Security est un projet du portfolio Spring qui propose une solution de sécurité complète intégrée aux systèmes utilisant Spring avec pour objectif de proposer un système complet de gestion de la sécurité. Le premier avantage de Spring Security est sa portabilité. Ne dépendant pas d'un serveur d'applications particulier, son utilisation est identique quel que soit le serveur utilisé. Cette portabilité est particulièrement importante si l'application développée doit pouvoir être vendue à un grand nombre de clients possédant des systèmes hétérogènes [28].

Le deuxième avantage de Spring Security est qu'il fournit en standard un nombre de fonctionnalités beaucoup plus important qu'un serveur Java EE classique.

Parmi les plus simples, et qui manquent cruellement dans la spécification Java EE, citons l'authentification automatique par cookie pour un nombre donné de jours, ainsi que la vérification qu'un utilisateur n'est pas déjà authentifié avec le login demandé [28].

Spring Security propose en outre des fonctionnalités avancées, telles que le support de solutions de Single Sign-On (une authentification unique pour l'ensemble des applications de l'entreprise) ou la sécurisation des objets de domaine, fournissant ainsi une aide considérable au développement d'applications ayant des besoins complexes en matière de sécurité [28].

Enfin, Spring Security permet de rendre la sécurité complètement transversale et déclarative. Une application peut pratiquement être développée entièrement sans se soucier de la sécurité, la configuration de celle-ci se faisant généralement dans un fichier dédié [28].

Spring Security apporte les avantages suivants :

• *Gestion des utilisateurs* **:** solution intégrée, complète et portable.

• *Sécurisation des requêtes HTTP* **:** disponible de manière plus fine que dans la norme Java EE.

• *Sécurisation de la couche de service* **:** API complète, qui s'intègre dans les Framework courants de POA.

• *Sécurisation de la couche de domaine* **:** listes de contrôle d'accès (ACL), qui peuvent être spécifiées au niveau de chaque objet de domaine.

• *Contrôle du code exécuté* : non pris en compte par Spring Security. Comme nous l'avons vu, cette fonction est rarement utile dans une application d'entreprise.

Conclusion

Ce chapitre nous a permis de faire un survol des principales failles de sécurité des applications web ainsi que leurs contre-mesures respectives telles que définies dans le top 10 de l'OWASP. En outre, il nous a permis de présenter les différentes solutions que nous allons utiliser lors de l'implémentation pour sécuriser notre système.

CHAPITRE 5 : IMPLEMENTATION DU NOUVEAU SYSTEME DE GESTION ET OPTIMISATION DU PROCESSUS DE RECENSEMENT PRE-ELECTORAL.

Dans ce chapitre, nous passons à la phase d'implémentation de notre système de gestion et optimisation du processus de recensement pré-électoral. Mais avant de passer à la présentation de certaines fonctionnalités implémentées, nous allons d'abord présenter l'architecture choisie (architecture client/serveur) pour implémenter notre système, les plateformes, les langages et les outils de développement.

Afin d'assurer un développement rapide, le nouveau logiciel va être développé sur la base d'une collection de Framework de développement, cela nous permettant de faire la programmation orientée composant. Un Framework est un ensemble de codes qui fournit une organisation ainsi qu'un grand nombre de fonctionnalités. Il fournit un socle solide avec une organisation bien définie ainsi qu'une multitude de classes et fonctions.

Dans l'architecture à trois niveaux, les applications au niveau serveur sont délocalisées, c'est à-dire que chaque serveur est spécialisé dans une tâche (serveur web/ serveur de base de données par exemple).

Il permet :

- Une plus grande flexibilité/souplesse ;
- Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau ;
- De meilleures performances, étant donné le partage des tâches entre les différents serveurs.

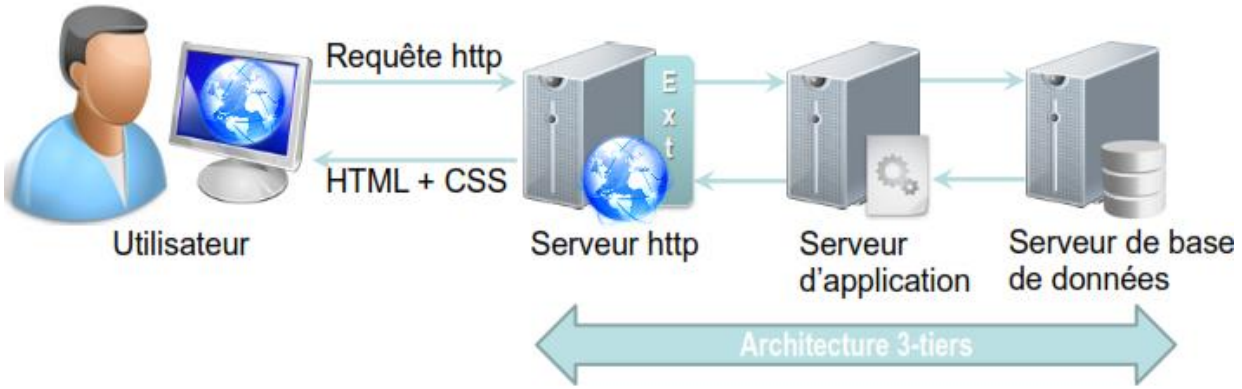


FIGURE 26: ARCHITECTURE 3-TIERS, CLIENT/SERVEUR

Cette architecture (appelée 3 tiers) fait intervenir trois parties indépendantes les unes des autres :

- **La couche de données liée au serveur de base de données (SGBD) :** stockage et accès aux données. Le système de stockage des données a pour but de conserver une quantité plus ou moins importante de données de façon structurée. Nous pouvons utiliser pour cette partie des systèmes très variés qui peuvent être des systèmes de fichiers, des mainframes, des systèmes de bases de données relationnelles, etc.
- **La logique applicative :** elle se compose généralement d'un script ou d'un programme qui constitue les traitements métier nécessaires sur l'information afin de le rendre exploitable par chaque utilisateur.
- **La couche présentation (ou affichage)** : elle est associé au client qui de fait est dit « léger » dans la mesure où il n'assume aucune fonction de traitement à la différence du modèle 2-tiers. C'est la partie la plus immédiatement visible pour l'utilisateur. Elle a donc une importance primordiale pour rendre l'information lisible, compréhensible et accessible.

V.1. Design pattern Modèle-Vue-Contrôleur (MVC)

Le concept de base du design pattern MVC repose sur la séparation des données et des vues dans des logiques différentes.

Le pattern MVC est basé sur trois couches :

Le Modèle décrit ou contient les données manipulées par l'application. Dans le cas typique d'une base de données le modèle offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle sont dénués de toute présentation [28].

La vue correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle qui lui sont passés par le contrôleur. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, soumission de formulaires). Ces différents événements sont envoyés au contrôleur. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle [28].

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ensuite avertit la vue que les données ont changé pour qu'elle se mette à jour [28].

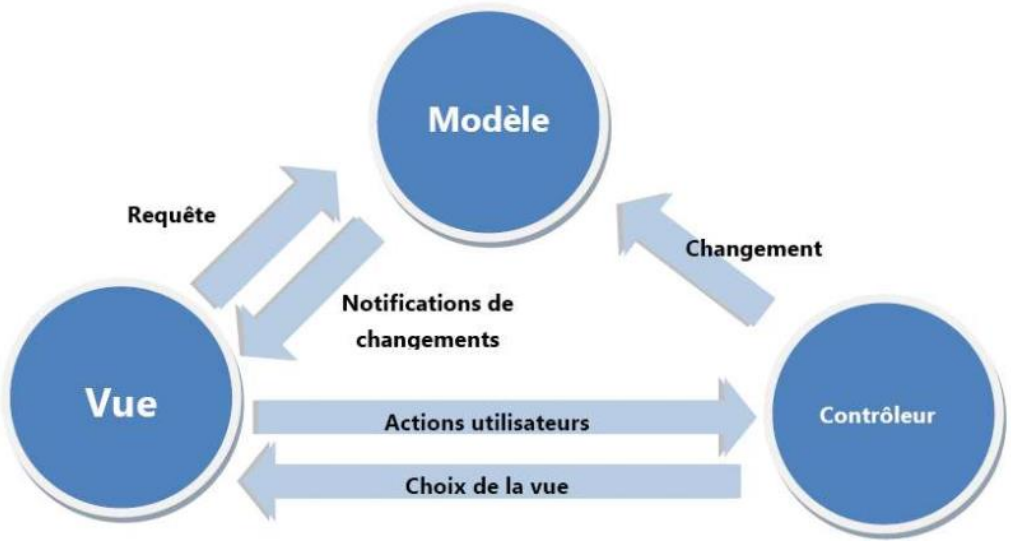


FIGURE 27: SCHEMA DE PRESENTATION DU MODELE MVC

En résumé, lorsqu'un client envoie une requête à l'application :

1. la requête est analysée par le contrôleur
2. le contrôleur demande au modèle approprié d'effectuer les traitements

3. le contrôleur renvoie la vue adaptée.

V.2. Plateformes, langages et technologies utilisés

Dans cette section, nous allons présenter les plateformes, langages ainsi que les technologies utilisées dans le développement de notre système.

1. Twitter Bootstrap

Twitter Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'applications web. C'est un ensemble qui contient des codes HTML 5 et CSS 3, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

2. Java EE

Java est un langage de programmation orienté objet, développé par Sun Microsystems et destiné à fonctionner dans une machine virtuelle, il permet de créer des logiciels compatibles avec des nombreux systèmes d'exploitation. Java EE (Java Enterprise Edition) est une plate-forme fondée sur le langage Java et son environnement d'exécution, qui regroupe un ensemble de technologies destinées à développer des applications d'entreprise. Ces applications ont la plupart du temps une composante serveur et sont concurrentes, c'est-à-dire qu'elles sont susceptibles d'être accédées par plusieurs utilisateurs simultanément.

3. Spring Framework

Spring est né de l'idée de fournir une solution plus simple et plus légère que celle proposée par Java 2 EE. C'est pour cette raison que Spring a été initialement désigné comme un conteneur léger (lightweight container). Son idée principale est de proposer un Framework qui utilise de simples POJO pour développer des applications plutôt que d'utiliser des EJB complexes dans un conteneur.

Spring propose de nombreuses fonctionnalités de base pour le développement d'applications :

- Un conteneur léger implémentant le design pattern IoC (Inversion of Control) pour la gestion des objets et de leurs dépendances en offrant des fonctionnalités avancées concernant la configuration et l'injection

automatique. Un de ses points forts est d'être non intrusif dans le code de l'application tout en permettant l'assemblage d'objets faiblement couplés.
- Une gestion des transactions par déclaration offrant une abstraction du gestionnaire de transactions sous-jacent.
- Faciliter le développement des DAO de la couche de persistance en utilisant JDBC, JPA, JDO ou une solution open source comme Hibernate, iBatis, ... et une hiérarchie d'exceptions.
- Un support pour un usage interne à Spring (notamment dans les transactions) ou personnalisé de l'AOP qui peut être mis en œuvre avec Spring AOP pour les objets gérés par le conteneur et/ou avec AspectJ
- Etc...

Ainsi, une application typique utilisant Spring est généralement structurée en trois couches dont la couche présentation constituée par l'interface homme machine, la couche service faite par interface métier avec mise en œuvre de certaines fonctionnalités (transactions, sécurité, ...) et enfin la couche accès aux données : recherche et persistance des objets du domaine. Spring MVC est un Framework basé sur Spring dont le but est le développement d'application web pour entreprise suivant le design pattern MVC.

4. Codename One Framework

Codename One est un framework open-source écrit en Java permettant de faire de la programmation mobile multiplateforme. Se présentant sous la forme d'un plug-in disponible pour les trois environnements de développement majeurs en Java (NetBeans, Eclipse et IntelliJ IDEA), il permet de cibler cinq plateformes mobiles (iOS, Android, Windows Phone, BlackBerry, J2ME) avec un code unique et a aussi pour particularité d'utiliser le cloud pour la compilation.

Le plug-in est composé de quatre parties majeures que voici :

- *Une API* : cette API contient toutes les classes nécessaires à la conception d'une application mobile et est écrite en langage Java.
- *Un designer* : fourni sous forme de logiciel, le designer permet de concevoir visuellement une interface d'application, de gérer la traduction d'une application en diverses langues, de créer des thèmes, de manipuler des images, etc.

- *Un simulateur* : il permet de tester ses applications sur son ordinateur. N'étant pas un émulateur, ce simulateur est rapide à l'exécution et embarque des outils pratiques pour tester en profondeur les applications.
- *Un serveur de compilation dans le cloud* : **c**e serveur permet de compiler en ligne les applications écrites avec Codename One.

Basé sur l'architecture de Java, Codename One est un ensemble d'outils pour le développement d'applications mobiles dont le but est de rendre plus facile le développement en unifiant la complexité de la programmation mobile en un ensemble d'outils, API et services ; ce qui amène l'ancien et fameux slogan de Java « Write Once Run Anywhere » (WORA) sur les terminaux mobiles.

V.3. Outils de développement

1. Environnement de développement intégré : Spring STS

Spring Tool Suite est un environnement de développement basé sur Eclipse qui est personnalisé pour développer des applications Spring. Il fournit un environnement prêt à utiliser pour implémenter, déboguer, exécuter et déployer vos applications Spring, y compris les intégrations pour Pivotal tc Server, Pivotal Cloud Foundry, Git, Maven, AspectJ, et vient au courant des dernières versions d'Eclipse.

2. Outil de gestion des dépendances : Apache Maven

Apache Maven est un outil pour la gestion et l'automatisation de production des projets logiciels Java en général et Java EE en particulier. L'objectif recherché est comparable au système make sous Unix : produire un logiciel à partir de ses sources, en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication [36].

3. Système de gestion de base de données : MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multithread et multi-utilisateur.

V.4. Serveurs d'application et proxy

1. Serveur d'application : Apache Tomcat 8.x

Apache Tomcat est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process3, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur http [37] [38].

2. Serveur proxy et WAF : NGiNX

Nginx est un serveur web open source développé pour répondre à une forte charge [un grand nombre de requêtes HTTP], grâce à une gestion des requêtes asynchrones. Ses points forts sont donc, une stabilité accrue, une faible consommation de mémoire, une simplicité de configuration, qui le rend agréable à administrer [45].

Nginx peut être configuré pour être un proxy inverse (en anglais : reverse proxy) Web et un serveur proxy de messagerie électronique (IMAP / POP3). L'utilisation la plus fréquente de Nginx est de le configurer comme un serveur Web classique pour servir des fichiers statiques et comme un proxy pour les requêtes dynamiques typiquement acheminées en utilisant une interface FastCGI vers un ou des serveurs applicatifs avec un mécanisme de répartition de charge [45].

Ainsi dans notre système, Nginx va non seulement nous servir de serveur proxy mais aussi de pare-feu applicatif (en anglais web application firewall : WAF) grâce au module complémentaire NAXSI.

V.5. Présentation de quelques fonctionnalités du système réalisé

Dans cette section, nous présenterons quelques captures d'écrans d'interfaces graphiques de notre nouveau système de gestion et optimisation du processus de recensement pré-électoral de la CENI.

1. Authentification

Cette interface permet aux utilisateurs de se connecter au système en saisissant leurs noms d'utilisateur et mot de passe.

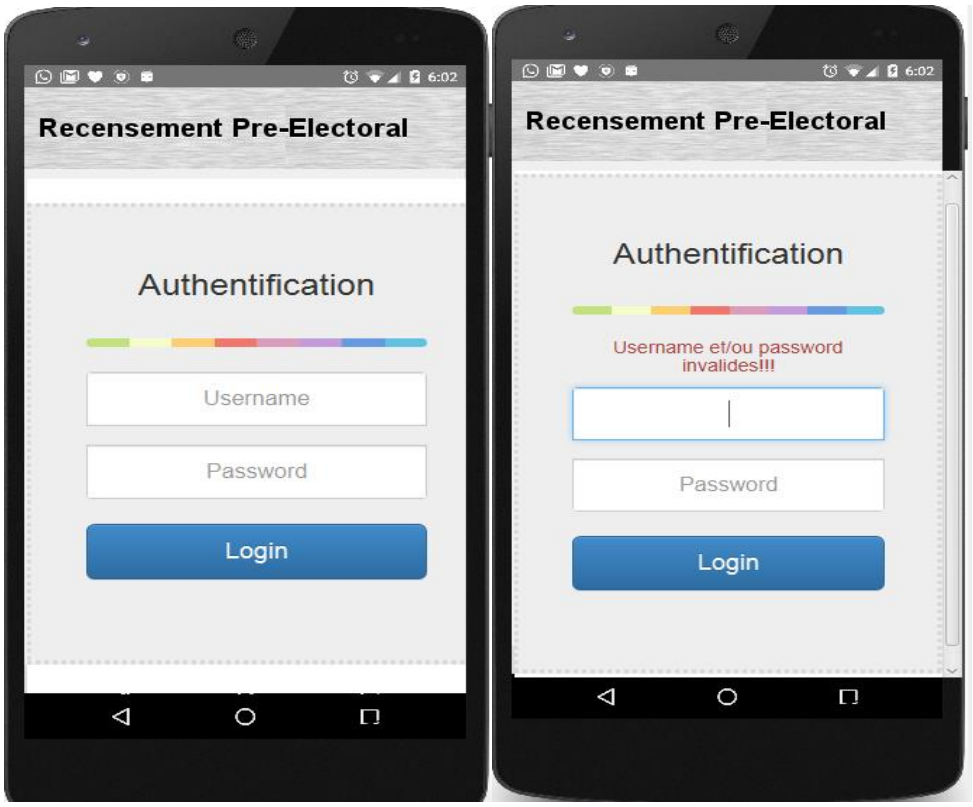


FIGURE 28: FORMULAIRES D'AUTHENTIFICATION

2. Inscription d'un candidat électeur

Cette interface permet aux agents recenseurs d'inscrire les candidats électeurs avec des terminaux mobiles.

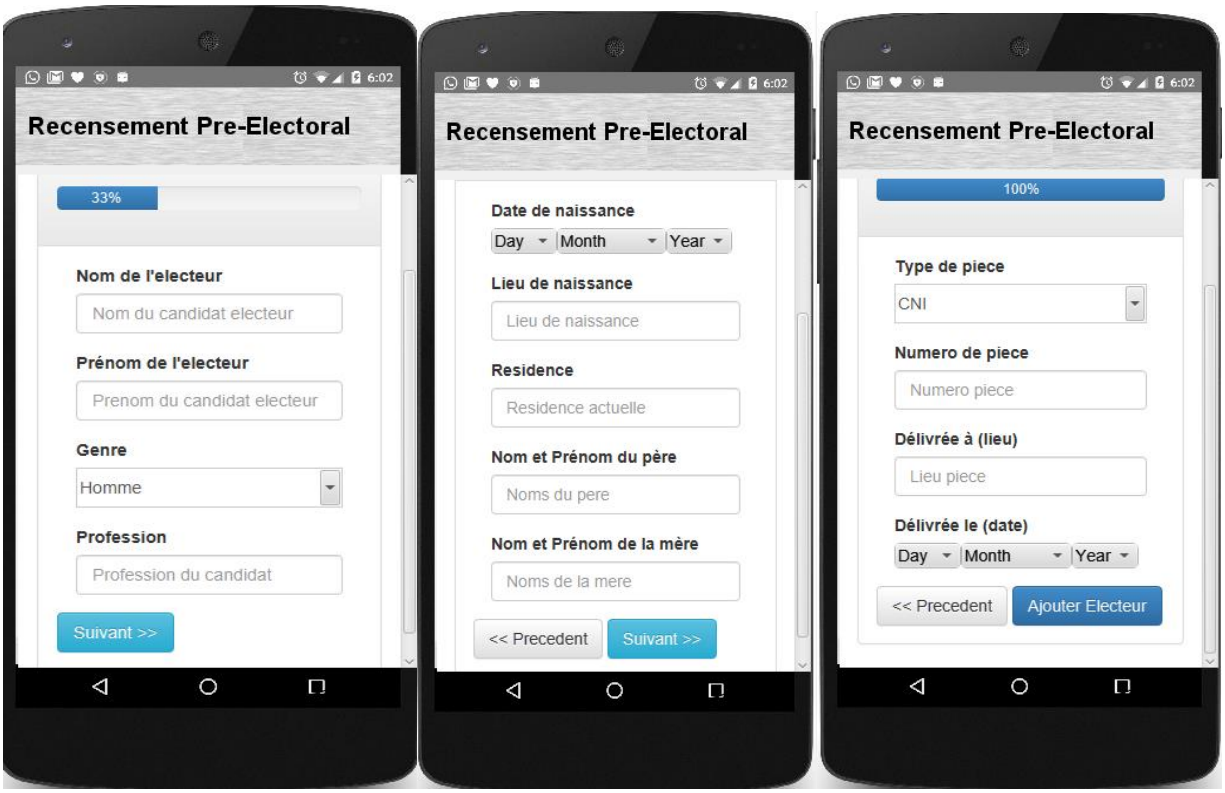


FIGURE 29: FORMULAIRES D'INSCRIPTION D'UN CANDIDAT ELECTEUR

3. Affichage du rapport d'activités individuel

Cette interface permet aux agents recenseurs d'afficher leurs rapports individuels global et historique par date.

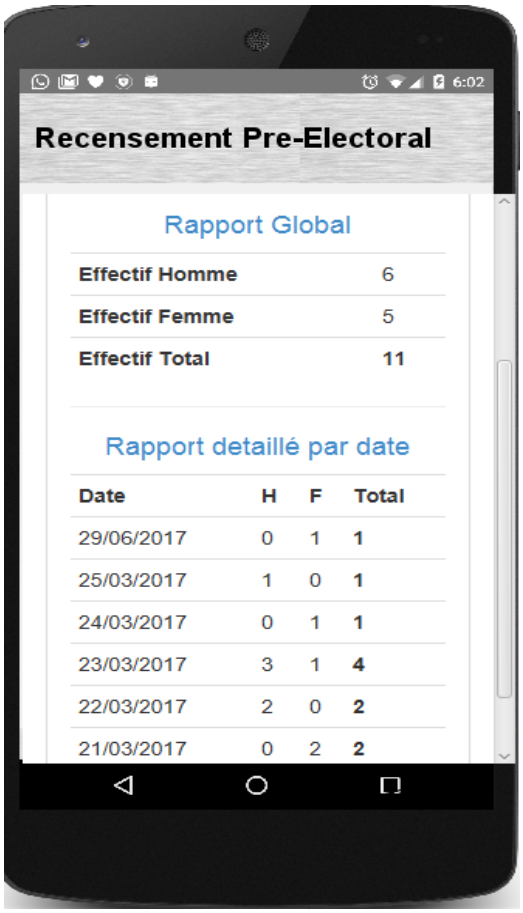


FIGURE 30: INTERFACE D'AFFICHAGE DU RAPPORT INDIVIDUEL D'UN AGENT RECENSEUR

4. Affichage du rapport d'activités du centre

Cette interface permet aux agents encadreur des centres d'inscription de visualiser le rapport global du centre, par date et enfin par agent recenseur. En plus, ces différents rapports sont détaillés et groupés par genres (Homme ou Femme).

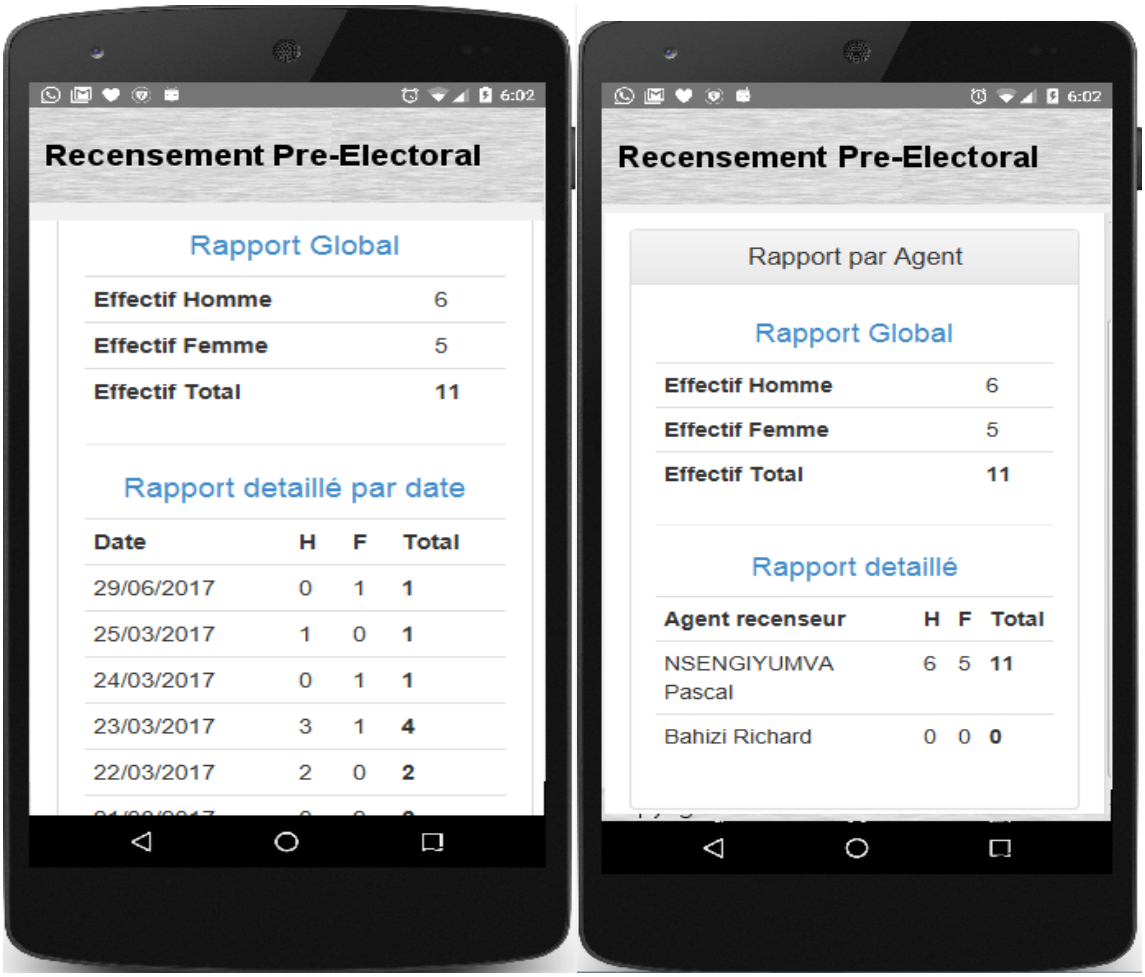


FIGURE 31: INTERFACE D'AFFICHAGE DU RAPPORT DU CENTRE D'INSCRIPTION

5. Affichage du rapport d'activités de la CECI

Cette interface permet aux membres de la CECI de consulter les rapports d'activités des différents centres d'inscription travaillant dans leurs CECIs. De même que pour les centres, les rapports sont synthétisés et groupés par genre, date et centre.

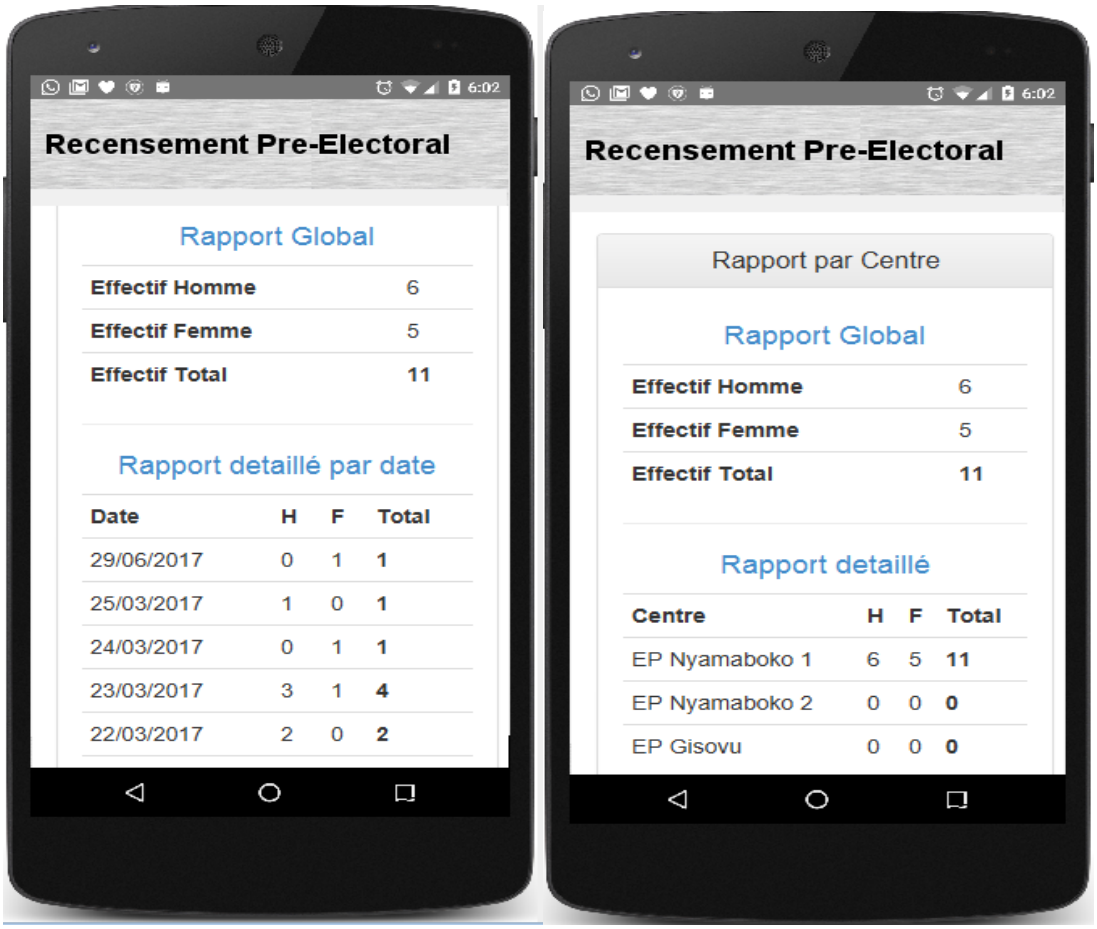


FIGURE 32: INTERFACE D'AFFICHAGE DU RAPPORT DE LA CECI

6. Affichage du rapport d'activités de la CEPI

Cette interface permet aux membres de la CEPI de visualiser les rapports des CECI de cette dernière, en les groupant par date, genre et CECI.

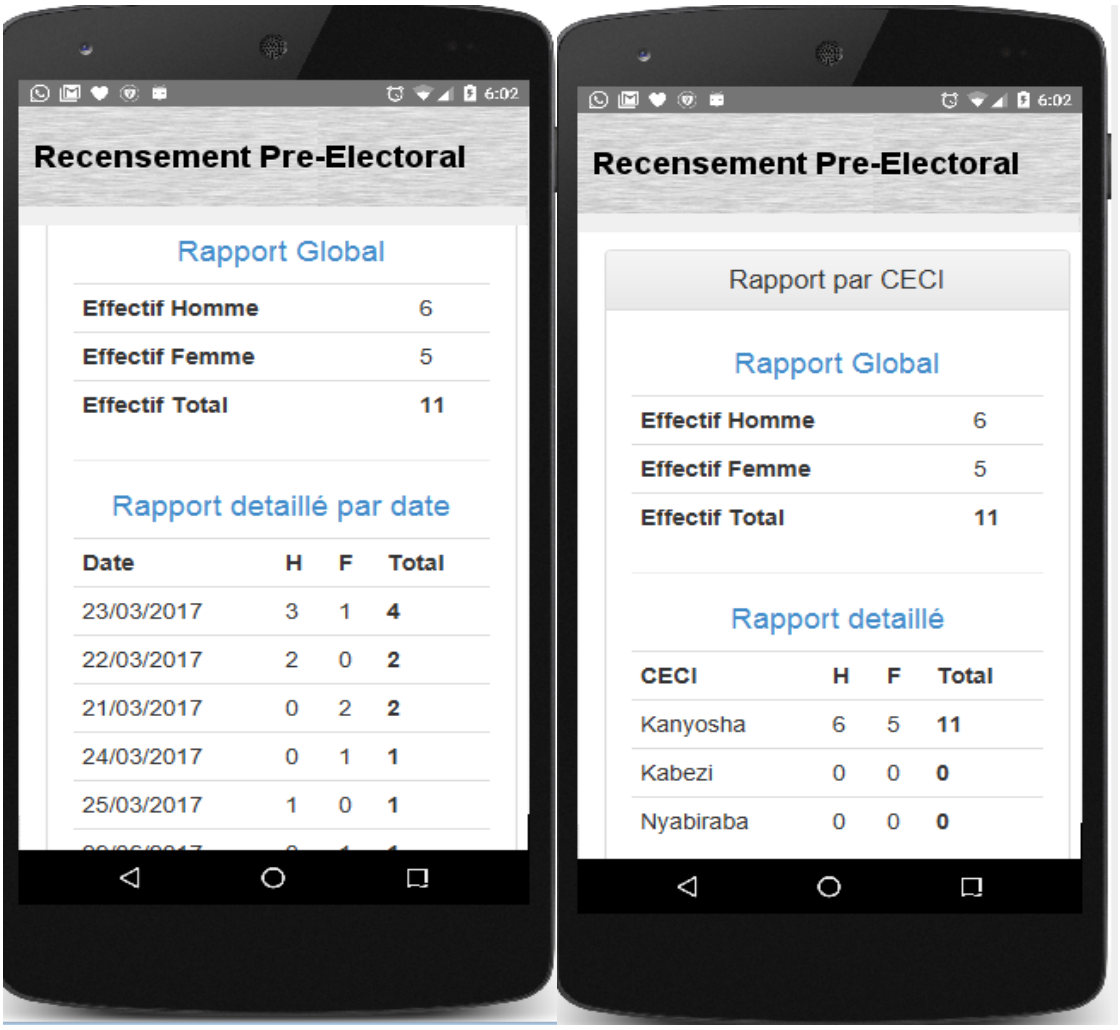


FIGURE 33: INTERFACE D'AFFICHAGE DU RAPPORT DE LA CEPI

7. Affichage du rapport d'activités de la CENI

Cette interface permet aux membres de la CENI de consulter les rapports des CEPIs.

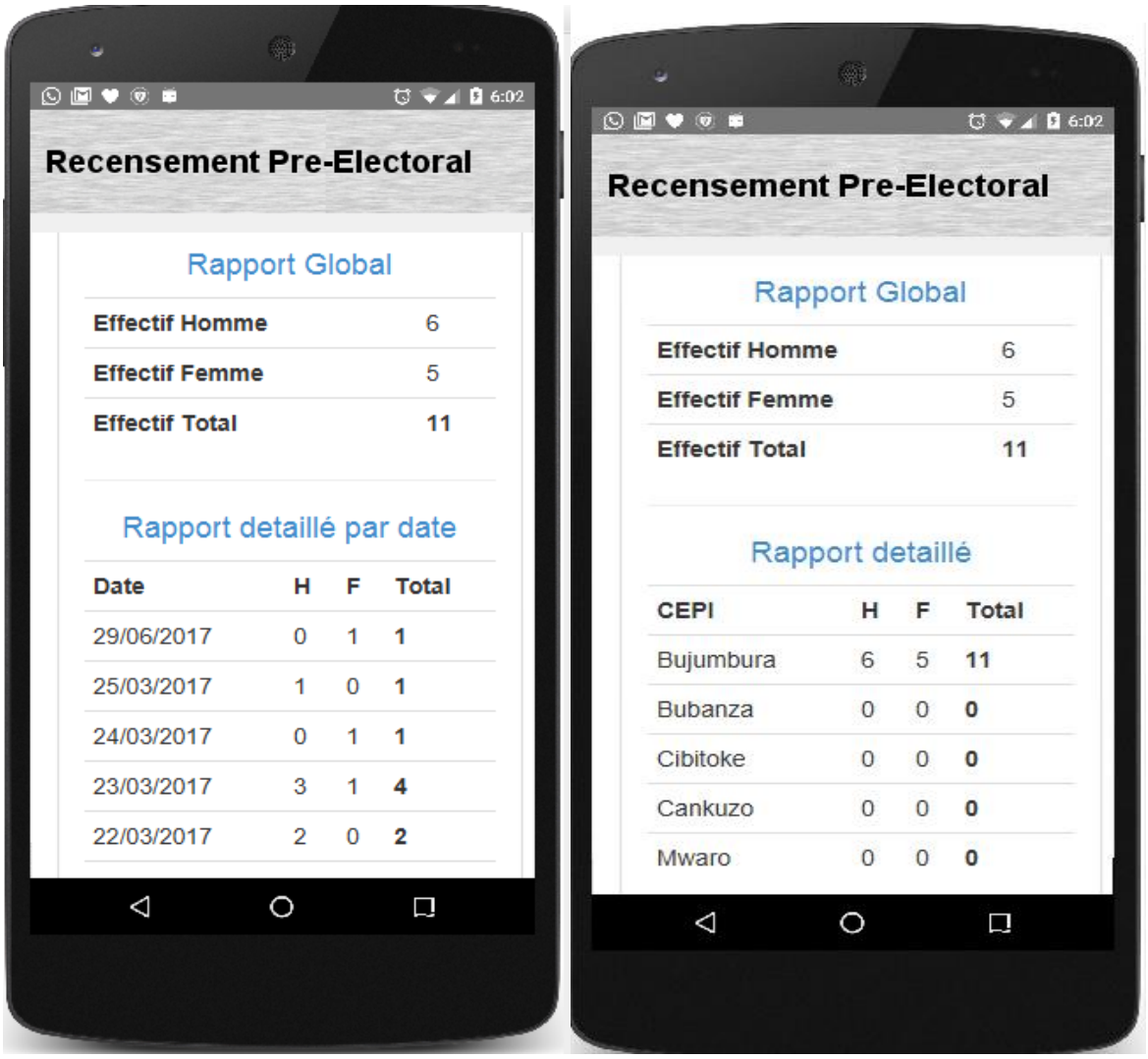


FIGURE 34: INTERFACE D'AFFICHAGE DU RAPPORT DE LA CENI

Conclusion

Ce chapitre nous a permis de définir les concepts généraux des applications d'entreprise entre autres l'architecture N-Tiers, le pattern MVC et les outils de développement. Nous avons parlé des plateformes et technologies utilisées pour implémenter notre système, et enfin nous avons passé à la présentation de quelques interfaces graphiques des fonctionnalités réalisées dans notre système.

CONCLUSIONS GENERALES

Au terme de notre projet de fin d'étude qui porte sur la conception et l'implémentation d'un système automatisé de gestion et optimisation du processus de recensement pré-électoral de la CENI, le moment est venu de dresser un bilan des résultats de notre recherche avant d'émettre et formuler des conclusions y relatives.

La mise en place de ce système a consisté en une série de phases entre autres le recueil des données, l'analyse et critique du système existant, conception et implémentation du nouveau système. En plus de ces phases, nous avons aussi fait une étude du comportement du nouveau système à l'aide de la fameuse théorie des files d'attente.

Le développement de ce système nous a permis de répondre aux problèmes évoqués dans la critique du système existant. Ainsi, notre système va améliorer considérablement la façon dont le processus de recensement pré-électoral est effectué actuellement.

Les fonctionnalités implémentées dans notre système :

- Inscription des candidats électeurs à partir des terminaux mobiles ;
- Établissement et visualisation des rapports du processus de recensement en temps réel et historisés, et en les groupant selon des critères de choix ;
- Gestion des centres d'inscription, des agents recenseurs… ;
- Génération des listes des candidats électeurs ;

permettent d'effectuer un suivi du processus de recensement pré-électoral en temps réel tout en optimisant sa durée afin de ne pas entraver les étapes suivantes du cycle électoral.

Enfin, ce travail nous a permis d'acquérir une expérience dans le développement des applications d'entreprise avec Java EE et particulièrement avec le Framework Spring MVC. C'était une bonne occasion pour confronter les connaissances théoriques apprises en les mettant au profit des entreprises – industrie dans la résolution de problèmes réels que ces dernières rencontrent.

RECOMMANDATIONS

Avant de clore notre projet, nous voudrions formuler des recommandations à l'endroit de la CENI d'une part et pour les futurs chercheurs et memorands d'autre part.

A la CENI, nous recommandons :

- de se doter d'un outil informatique approprié, et en suffisance afin de pouvoir déployer et exploiter le système automatisé de gestion et optimisation du processus de recensement pré-électoral;
- d'adopter le système automatisé de gestion et optimisation du processus de recensement pré-électoral dans le recensement pré-électoral venir afin d'optimiser la durée du processus et aussi minimiser les dépenses dues à l'achat des registres d'inscription et d'autres documents connexes;
- d'organiser des séminaires et formations à l'endroit de tous les acteurs impliqués dans le processus de recensement en particulier et du cycle électoral en général;
- de développer des modules complémentaires afin d'automatiser toutes les phases du cycle électoral entre autres le recensement, la gestion des observateurs et mandataires, la gestion des candidats aux élections, la gestion des élections, la publication des résultats en ligne,…

Aux futurs chercheurs et memorands :

- de poursuivre des recherches sur ce projet en ajoutant d'éventuelles fonctionnalités qui seraient jugées nécessaires dans le but de répondre aux besoins et attentes des utilisateurs, et ceci dans le but d'améliorer les fonctionnalités actuelles;
- de développer des modules additionnels entre autres le module de gestion des observateurs et mandataires, module de messagerie et forum intégrés, ainsi que d'autres modules de gestion et automatisation des autres processus connexes au processus de recensement;
- d'améliorer la sécurité de notre système en y intégrant un module basé sur la biométrie (reconnaissance faciale, reconnaissance de l'ADN, …) pour contrer toute tentative de fraude dans le fichier électoral.

RÉFÉRENCES BIBLIOGRAPHIQUES ET WEBOGRAPHIQUES

[1] Décret n°100/22 du 20 février 2009 portant organisation et fonctionnement de la CENI

[2] Loi n°1/010 du 18 mars 2005 portant promulgation de la constitution de la république du Burundi

[3] Rapport général du processus électoral de 2015

[4] Loi n° 1/ 20 du 3 juin 2014 portant révision de la loi n°1/22 du 18 septembre 2009 portant code électoral

[5] Ian Sommerville, Software engineering (9th Edition), Addison-Wesley 2010

[6] Jim Arlow and Ila Neustadt, UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design Second Edition, Addison-Wesley 2005

[7] Pascal Roques, UML 2 Par la pratique : Etude de cas et exercices corrigés 5e Edition, Eyrolles Septembre 2006

[8] Paule-Annick DAVOINE, Modélisation UML et outils pour l'analyse spatiale, LSR-IMAG, équipe SIGMA, Institut Polytechnique de Grenoble, Février 2001

[9] David Autissier et Valerie Delaye, Mesurer la performance du système d'information, Eyrolles Juin 2008

[10] Nick Rozanski and Eoin Woods, Software systems architecture, working with stakeholders using viewpoints and perspectives 2nd Edition, Addison-Wesley, 2005, 2012

[11] Robert Hanmer, Pattern-Oriented software architecture for dummies, Wiley Janvier 2013

[12] Thomas Erl, Service Oriented Architecture, Analysis and Design for services and microservices 2nd Edition, PRENTICE HALL ServiceTech PRESS Décembre 2016

[13] Tout sur la sécurité informatique 3e Edition, Dunod Paris 2005, 2009, 2013

[14] Guillaume HARRY, Failles de sécurité des applications Web : Principe, parades et bonnes pratiques de développement (Juillet 2012)

[15] Les dix risques de sécurité applicatives web les plus critiques, OWASP Top 10 - 2013 release

[16] Patrick Leclerc, Stratégies OWASP pour développer et exploiter des applications sécuritaires, The OWASP Foundation, ASIQ Mars 2016

[17] Sécurité des applications web : comment maitriser les risques liés à la sécurité des applications web ? CLUSIF Septembre 2009

[18] Matthieu Guionnet, NAXSI : un pare-feu applicatif pour NGINX [article]

[19] Guide de sécurité des applications web, Royaume du Maroc- Administration de la Défense Nationale - Direction Générale de la Sécurité des Systèmes d'Information

[20] Yakov Fain, Victor Rasputnis, Anatole Tartakovsky and Victor Gamov, Enterprise web development-Building HTML 5 applications: From Desktop to Mobile, O'Reilly Media Juillet 2014

[21] Jean Francois Nogier,Thierry Bouillot et Jules Leclerc, Ergonomie des interfaces, 5e Edition, DUNOD Octobre 2011

[22] Shai Almog, Chen Fishbein et Steve Hannah, Codename One Developer guide: Build NATIVE iOS, Android & Windows apps using Java, Amazon Digital Services LLC July 31, 2016

[23] Eric Dodji Gbofu, Codename One – Développer en Java pour iOS, Android, BlackBerry et Windows Phone, Éditions D-BookeR 2015

[24] Antonio Goncalves, Beginning Java EE 7, Apress Juin 2013

[25] Joseph Ottinger,Srinivas Guruzu,and Gary Mak, Hibernate Recipes, A problem solution approach (2nd Edition), Apress 2015

[26] Geoffroy Warin, Mastering Spring MVC 4, Packt, Septembre 2015

[27] Shameer Kunjumohamed, Hamidreza Sattari, Alex Brettet and Geoffroy Warin, Spring MVC: Designing Real-World Web Applications, Packt, Novembre 2016

[28] Arnaud Cogoluègnes,Thierry Templier,Julien Dubois,Jean-Philippe Retaillé, Spring par la pratique 2e Edition, Eyrolles, Juillet 2011

[29] Gary Mak, Spring par l'exemple, PEARSON, Avril 2009

[30] Yves Caumel, Probabilités et processus stochastiques, SPRINGER, Mai 2011

[31] Howard Anton and Chris Rorres, Elementary linear algebra, applications version Tenth Edition, John Wiley & Sons, Inc. 2010

[32] http://www.lesechos.fr/idees-debats/cercle/cercle-131863-les-7-peches-capitaux-en-matiere-de-securite-des-applications-mobiles-1111082.php consulté le 09/04/2017

[33] http://www.lemagit.fr/conseil/Securite-des-applications-mobiles-un-enjeu-difficile-de-lere-BYOD-mais-pas-une-cause-perdue consulté le 08/04/2017

[34] https://fr.wikipedia.org/wiki/Cycle_de_développement_(logiciel) consulté le 10/04/2017

[35] https://openclassrooms.com/courses/creez-des-applications-de-qualite-avec-le-design-pattern-mvc/le-genie-logiciel-gl consulté le 09/04/2017

[36] https://fr.wikipedia.org/wiki/Apache_Maven consulté le 10/04/2017

[37] https://fr.wikipedia.org/wiki/Apache_Tomcat consulté le 10/04/2017

[38] https://www.jmdoudoux.fr/java/dej/chap-tomcat.htm consulté le 11/04/2017

[39] https://fr.wikipedia.org/wiki/File_d'attente consulté le 10/04/2017

[40] http://www.sebsauvage.net/comprendre/ssl/ consulté le 10/04/2017

[41] https://thawte.tbs-ertificats.com/guides/Thawte_Understanding_SSL_Certificates_fr_FR.pdf consulté le 10/04/2017

[42] https://fr.wikipedia.org/wiki/Transport_Layer_Security consulté le 10/04/2017

[43] https://fr.wikipedia.org/wiki/HyperText_Transfer_Protocol_Secure consulté le 10/04/2017

[44] http://blog.sedona.fr/2013/03/top-10-owasp-2013-les-failles-de-securite-web-les-plus-courantes/ consulté le 26/06/2017

[45] https://lindev.fr/?post/2010/04/02/Nginx-Premiers-pas consulté le 12/04/2017