



République du Sénégal

Un Peuple-Un But-Une Foi

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



Université Iba Der Thiam de THIES
UFR sciences et Technologies

Filière Informatique option Réseaux et Télécommunications
Niveau : Master I

Projet : Service Web

Gestion et Suivi des Résultats des Elections

Présenté par :

Baye Ousseynou FALL

Ahmed LO

Professeur :

Pr M. THIAM

Année académique 2021 – 2022

Plan de travail

Introduction

I. Objectif

II. Etude préliminaire

a) Identification des acteurs

b) Identification des messages

c) Modélisation du Contexte

III. Modélisation

a) Diagramme de Cas d'utilisation

b) Diagramme de Classe

c) Diagramme de Séquence

IV. Choix Technologique

V. Implémentation et Test de l'application

Conclusion

Introduction

Pour le renforcement des capacités d'un état en matière d'élections et de référendums, permettant ainsi l'acquisition à temps de la logistique, une bonne organisation d'élection avec des résultats incontestables participerait à une démocratie inclusive conduisant à un climat favorable à l'environnement des affaires.

Dans l'optique d'une telle organisation électorale, nous allons mettre en place un système de gestion et de suivi des résultats des élections. Ce système couvre les élections de la phase d'inscription des candidats et des électeurs à la sortie des résultats officiels en passant par le dépouillement et les éventuelles réclamations.

I. OBJECTIFS

L'objectif de ce travail consiste à mettre en place une plateforme permettant aux électeurs de voter en ligne. De ce fait nous serons chargés de modéliser une application en s'appuyant sur les aspects de sécurité et de gestion du système.

- En ce qui concerne l'aspect sécurité, il donne la possibilité à chaque usager votant de s'inscrire à une élection suivant une politique de sécurité bien définie (si son inscription est valide, l'utilisateur recevra un message qui lui indique son login et son mot de passe par défaut). En ce qui concerne l'administrateur du système, il est chargé d'enregistrer les listes électorales, de paramétrer le système de vote, de gérer les comptes des utilisateurs et son compte lui-même.
- Pour la gestion du système, elle donne la possibilité d'afficher les listes électorales, le nombre d'électeur ayant participé ou non aux élections et de donner le pourcentage de vote obtenue pour chaque liste électorale.

II. Etude préliminaire

L'étude préliminaire est la toute première étape de notre processus de développement. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement des textes illustratifs et des diagrammes très simples.

Elle prépare les activités plus formelles de capture des besoins fonctionnels et de capture des besoins techniques.

a) Identification des acteurs

Après analyse du cahier des charges nous avons pu identifier les principaux acteurs du système à développer. A savoir

- User : c'est l'utilisateur général du système.
- L'Electeur : un électeur doit d'abord s'inscrire sur la plateforme pour pouvoir voter sur une liste électorale de son choix.
- Représentant : c'est un électeur chargé de représenter une liste électorale.
- L'Administrateur : l'administrateur est chargé d'administrer le système, gérer les différents comptes des utilisateurs et de gérer les listes électorales.

b) Identification des services

➤ Electeur :

- Inscrire sur la plateforme
- Vérifier son inscription (service)
- Demande de changement de bureau de vote (service)
- Voter (service)
- Consulter les résultats (service)

➤ Administrateur :

- Permet aux électeurs de s'inscrire
- Ajouter et supprimer les listes électorales
- Afficher les listes électorales
- Valider le compte des utilisateurs
- Mettre à jour les résultats
- Vérification double inscription (service)
- Mettre à jour les résultats
 - Circonscriptions
 - Sous-circonscriptions

III. Modélisation

a) Diagramme de Cas d'utilisation

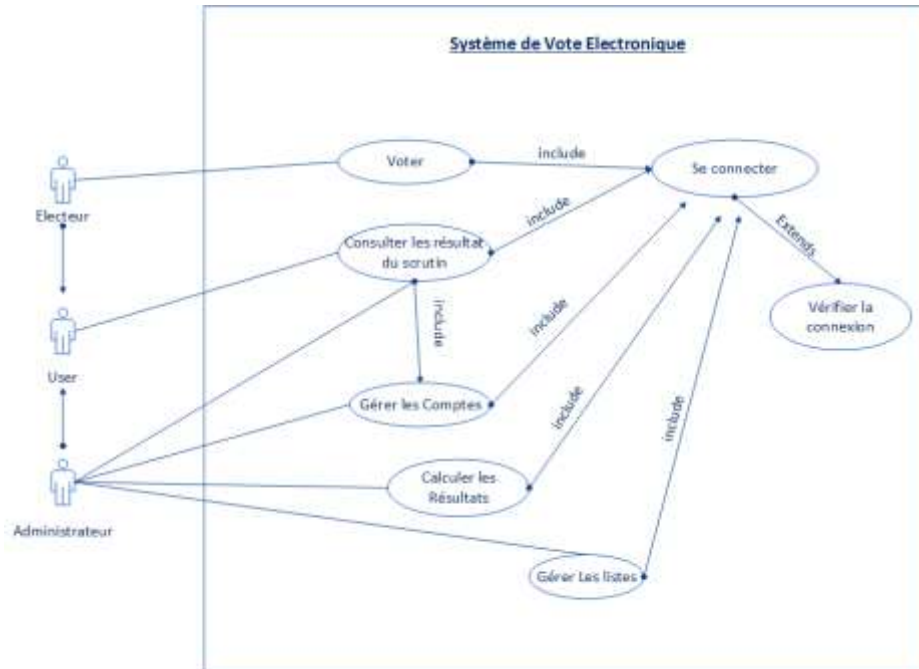


FIGURE 1 – Diagramme de Cas D'Utilisation

Lors de l'utilisation de l'application de vote, trois acteurs seront nécessaires. Tout d'abord, un administrateur qui ne peut pas voter et qui gère les scrutins ainsi que les comptes des utilisateurs. Ensuite deux acteur votants qui représenteront le coter client. Enfin les actions qui pourront être réalisées par ces deux derniers seront réalisées par l'utilisateurs général (User).

Détaillons les différentes actions que peuvent réaliser ces acteurs :

Administrateur :

- Gérer les Comptes : suppression ou création d'un nouvel votant et mise à jour la base de données.
- Calculer résultat : clos le scrutin et permet aux votants de voir les résultats.
- Réinitialiser referendum : permet de réinitialiser les votes de tous les votants et de créer un nouveau scrutin.

Electeur :

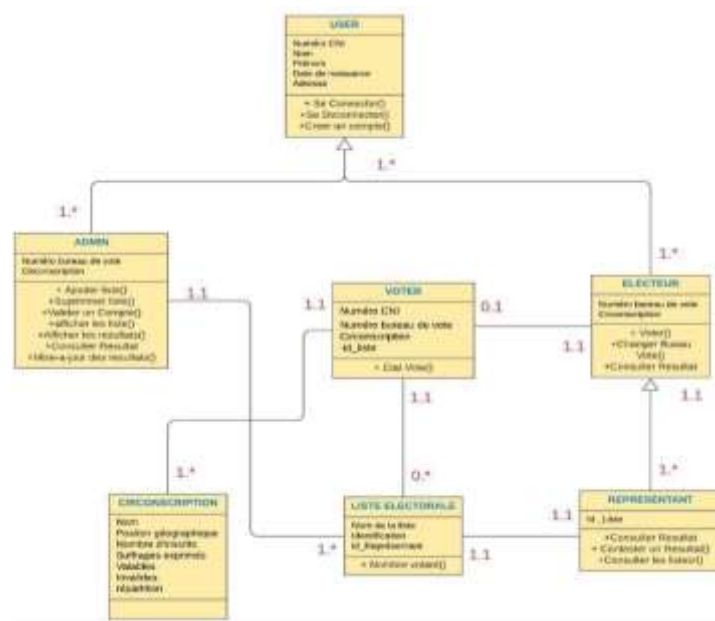
- Voter : le votant fait un choix parmi les listes qui lui sont Proposées.

User :

- Voir les votes réalisés : pour montrer le fonctionnement de l'application et assurer sa fiabilité, les utilisateurs pourront voir comment ont été enregistrés les votes dans la base de données.
- Consulter résultats : Le résultat final est déchiffré. Après cette action les votants ont accès au résultat du scrutin.
- Se connecter : l'utilisateur envoie ses identifiants afin de pouvoir accéder aux différentes fonctionnalités de l'application, selon son rôle : Electeur ou administrateur.
- Vérifier connexion : protocole réalisé par le serveur afin de vérifier les identifiants du votant ou de vérifier l'authentification de l'administrateur.

b) Diagramme de Classe

FIGURE 2 – Diagramme de Classe



Ce diagramme permet de présenter les différentes classes contenues dans notre application. Notre programme fonctionne avec un système client-serveur. Chaque classe d'objet contient des fonctionnalités et méthodes qui sont actionnées par des boutons. Certaines fonctionnalités ne sont disponibles qu'en fonction de l'utilisateur en question.

De plus du profile électeur donnée par le cahier des charges, nous avons décidé d'avoir Admin et User comme deux classes distinctes.

User vas représenter notre classe mère qui étant les classe Electeur et Admin.

c) Diagramme de Séquence

Dans une optique de simplification, on supposera que le serveur a été lancé en amont, l'exécution des fonctions nécessite préalablement de s'être connecté au service.

- **Diagramme de Séquence de la connexion**

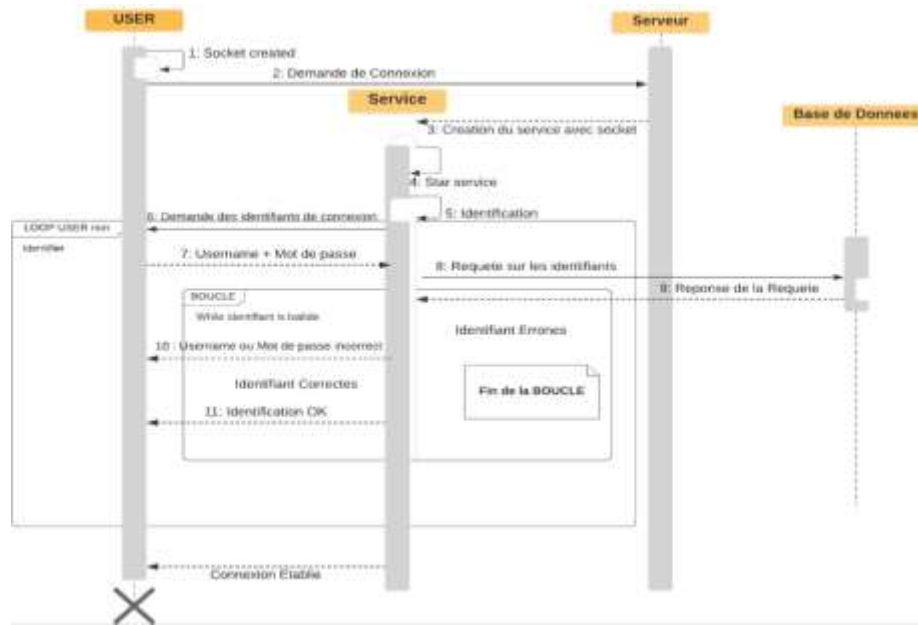


FIGURE 3 – Diagramme de Séquence de Connexion

Ce premier de séquence diagramme explicite les différentes étapes de la connexion d'un client (classe Electeur) au système(serveur). La communication entre le serveur et le client s'effectue via une instance de la classe Socket. Le client commence donc par créer son propre socket, puis se connecte au serveur via ce socket.

Le serveur est dit « multithread » : pour chaque Electeur qui se connecte, il va créer un processus (ou thread) avec lequel interagira l'électeur. Ce thread est modélisé ici par la classe Service. Le serveur crée donc le service, en lui passant en argument le socket avec lequel s'est connecté le votant. Une fois créé, le service lance le processus d'identification, correspondant à la boucle sur le diagramme. Le service demande au votant un identifiant (user Name) et un mot de passe, puis vérifie dans la base de données si un couple identifiant / mot de passe correspond à celui fourni par l'utilisateur. Tant qu'il ne trouve aucune correspondance, le service continue de demander au votant ses identifiants.

Une fois une correspondance trouvée, le service met fin à la boucle et se redirige l'électeur sur son tableau de bord.

- **Diagramme de Séquence de vote**

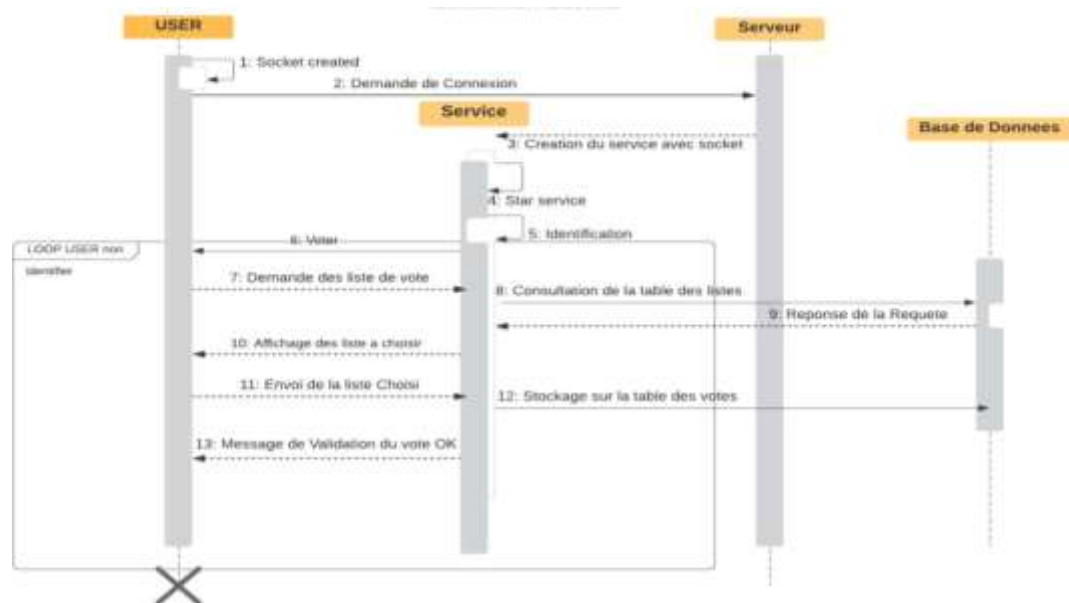


FIGURE 4 – Diagramme de Séquence de Vote

Ce diagramme détaille le processus de vote. Comme toujours, tout part d'une demande de l'utilisateur : la classe Electeur envoie une demande de vote au service via son socket. Le service récupère alors auprès du serveur le générateur et les listes électorales disponibles les transmet au votant. Ces informations sont stockées dans le serveur, puisqu'elles ne varient pas. Le votant sélectionne alors la liste de son choix et le transmet au service, qui va se charger de mettre à jour la base de données avec le nouveau vote.

- **Diagramme de Séquence de de Déconnexion**

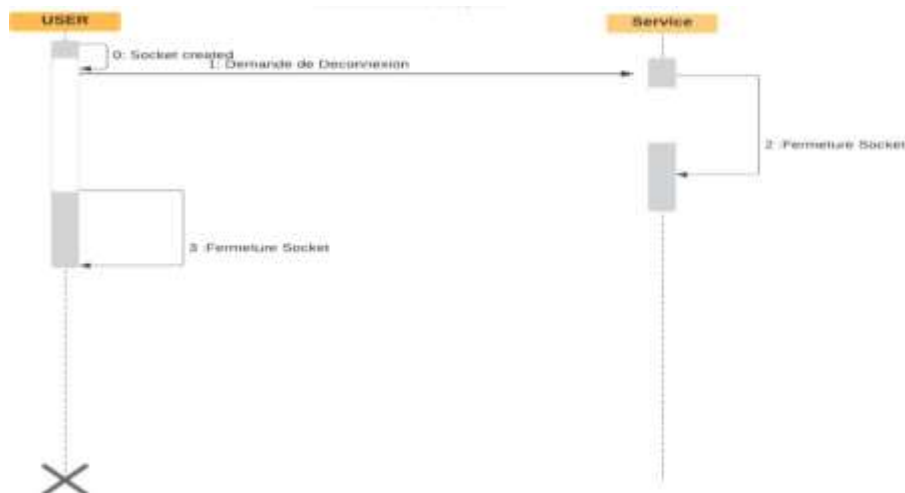


FIGURE 5 – Diagramme de Séquence de Déconnexion

Il reste enfin à présenter le processus de déconnexion. Ce processus est très simple : il consiste en la fermeture des deux sockets, celui du client (ici, l'électeur User) et celui du serveur (ici, le service). Cette fermeture est indispensable pour libérer les ressources utilisées par le socket. Une fois son socket fermé, le service va mettre fin à son exécution.

VI. Choix Technologique

Pour la réalisation de ce projet, nous avons utilisé différentes technologies à savoirs :

Eclipse IDE : Qui est un environnement de développement s'adaptant aux langages de programmation (JavaScript, PHP, Java EE, etc.).



Java : Nous avons choisi java comme langage de base pour implémenter ce projet. Ce langage nous a permis de générer nos différentes classes nécessaires pour réaliser les différentes missions renseignées sur le cahier des charges de référence.



Java EE : Qui est une plate-forme fortement oriente Serveur pour le développement et l'exécution d'applications web. C'est un langage de communication oriente serveurs qui nous a permis d'exécuter des requêtes utilisateurs depuis des feuilles JSP.



MySQL : Nous avons choisi MySQL comme SGBD pour gérer la connexion et les requêtes coté serveur vers notre base de données. Elle nous permet via le site d'exécuter des requêtes pour afficher les informations demandées au niveau de l'interface d'utilisateur courant.



Apache Tomcat v9 : Est un serveur internet, qui permet d'héberger des applications web code en Java EE.



VII. Implémentation ET Test de l'application

1) Implémentation

- Installer le serveur Tomcat sur Eclipse
- Créer le projet dans Eclipse
- La partie Backend dans le dossier « election », on crée les dossiers :
 - DAO (l'ensemble des fichiers qui gèrent les accès à la database)
 - Model (pour définir les modèles d'objet)
 - Web (fichiers servlet qui gèrent les intégrations)
- La partie Frontend dans « webapp »
 - Html (les fichiers html qui représentent les pages web)
 - Images (les images utilisées dans les html)
 - Js (fichiers qui gèrent l'aspect interactif dans la partie frontend)
 - Styles : contenant les Css (la présentation des documents html), bootstrap, etc.
 - WEB-INF : contient le dossier des librairies appelées
 - Les fichiers JSP : pour générer des pages web dynamiques

2) Visualisation des Différents interface de notre Application

Présentation de l'interface d'accueil

Le Capture ci-dessus représente la page d'accueil de notre application.

Avec cette interface, on donne la main au visiteur à travers un bouton de connexion qui lui permet de s'authentifier afin de pouvoir voter pour la liste de son choix.



Présentation de l'interface d'authentification

Cette Interface permet à un utilisateur possédant un compte de pouvoir se connecter en renseignant son Username et Password

Bienvenue Connectez vous

Enter Votre Cni...

Votre mot de passe

☐ Se Souvenir de Moi

[mot de passe oublié?](#)

[Login](#)

[Besoin d'un compte? Créer un compte!](#)

Copyright © gestion election 2022 [gestion - Terms & Conditions](#)

Présentation de l'interface d'inscription

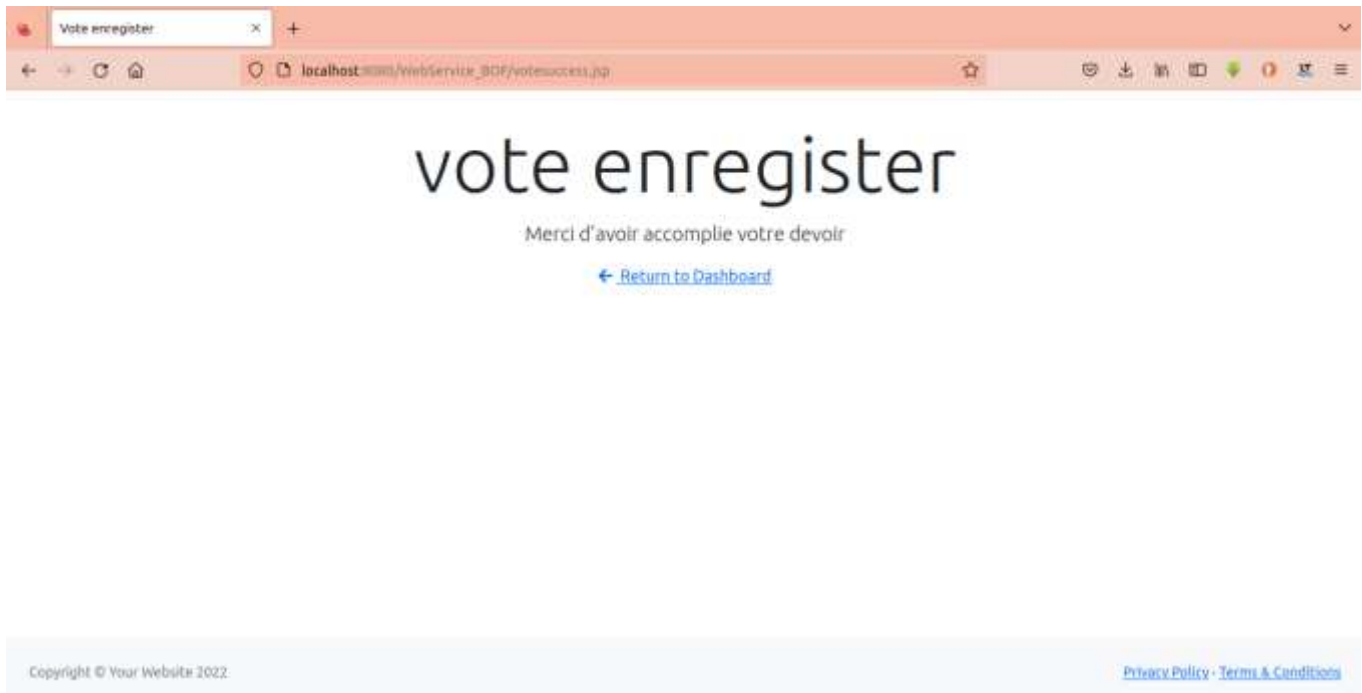
Cette Interface permet à un utilisateur ne possédant pas de compte de pouvoir en créer un, en remplissant le formulaire proposé.

A screenshot of a web browser displaying a form titled "Creation Compte". The form is centered on a blue background. It contains several input fields: "PRENOM", "NOM", "Numero cni", and "DATE DE NAISSANCE" (with a date format hint "jj / mm / aaaa"). There are also five dropdown menus labeled "Region", "Departement", "Communes", "Lieu de Vote", and "Bureau de Vote". At the bottom, there are two text boxes for "Votre mot de passe" and "Confirmer mot de passe", followed by a blue button labeled "s'inscrire". The browser's address bar shows "localhost:8080/webService_BOT/Register.jsp".

Présentation de l'interface voter

Après avoirs passer la procédure d'authentification, l'utilisateur pourras voter en sélectionnant juste une liste parmi les listes électorales disponible sur la base de données.

A screenshot of a web browser displaying a voting interface. The browser's address bar shows "localhost:8080/webService_BOT/voter.jsp". The interface consists of a single dropdown menu showing "MackySALL" with a downward arrow. Below the dropdown is a blue button labeled "Voter".



Présentation de l'interface affichage des résultats par circonscription

L'administrateur du système a la possibilité d'afficher les résultats provisoires des élections par circonscription.



Conclusion

Le système de gestion et de suivi des élections est réalisé avec des services pris en compte (Inscription sur la plateforme, connexion à son compte utilisateur, consulter la liste des candidats, voter, suivre les résultats, etc.) pour une bonne organisation des élections. Toutefois ce projet ne saurait être conclu définitivement sachant qu'il reste quand-même des services à ajouter, un niveau de sécurité à améliorer etc.

Ceci est donc un prototype du projet dans sa globalité et donc un travail complémentaire est à faire pour une meilleure version de ce système de gestion d'élection.