

Architecture du logiciel

Rédigé par :

- Belkada Merzouk
- Ralaïarisoa Ranto Andrianjohany
- Mohamed Salou Nabé

I. Interprétation du sujet

Ce qui est demandé dans ce projet est de développer un système capable de recréer l'environnement de vote pendant les élections. Ce dernier devra comprendre l'organisation de l'élection elle-même jusqu'au déroulement de l'élection.

Du fait qu'on parle d'élection, il faut donc que le système soit fiable, sûr, robuste et efficace. Le processus de vote doit être simple mais efficace. Il faut qu'il soit dénué de toutes fonctionnalités qui sont susceptible de compliquer l'utilisation de l'application.

II. Concepts

Electeurs :

Les électeurs seront les principaux utilisateurs du système. Ils peuvent voter selon le système électoral choisi pour une élection donnée

Elections :

Les élections seront organisées par des organisateurs (utilisateur du système) selon des règles bien définies comme le mode scrutin, le type d'élection, la date, l'ajout des candidats

Organisateurs :

Les organisateurs sont des utilisateurs du système. Ce sont eux qui organisent les élections et veillent au bon déroulement des élections

Candidats :

Les candidats sont des personnes qui se présentent à une élection donnée. Ils sont ajoutés par les organisateurs dans les élections pour être éventuellement choisis.

Scrutin :

Le scrutin permet le passage du décompte des voix à la désignation des élus. Le scrutin définit les règles à suivre pour le décompte afin de désigner le gagnant d'une élection donnée

Division territoriale :

Le système assimilera les divisions territoriales de la France. Entre autre les régions, les départements, les communes, les cantons et les circonscriptions.

Un électeur doit appartenir à une circonscription et à une seule.

Type d'élection :

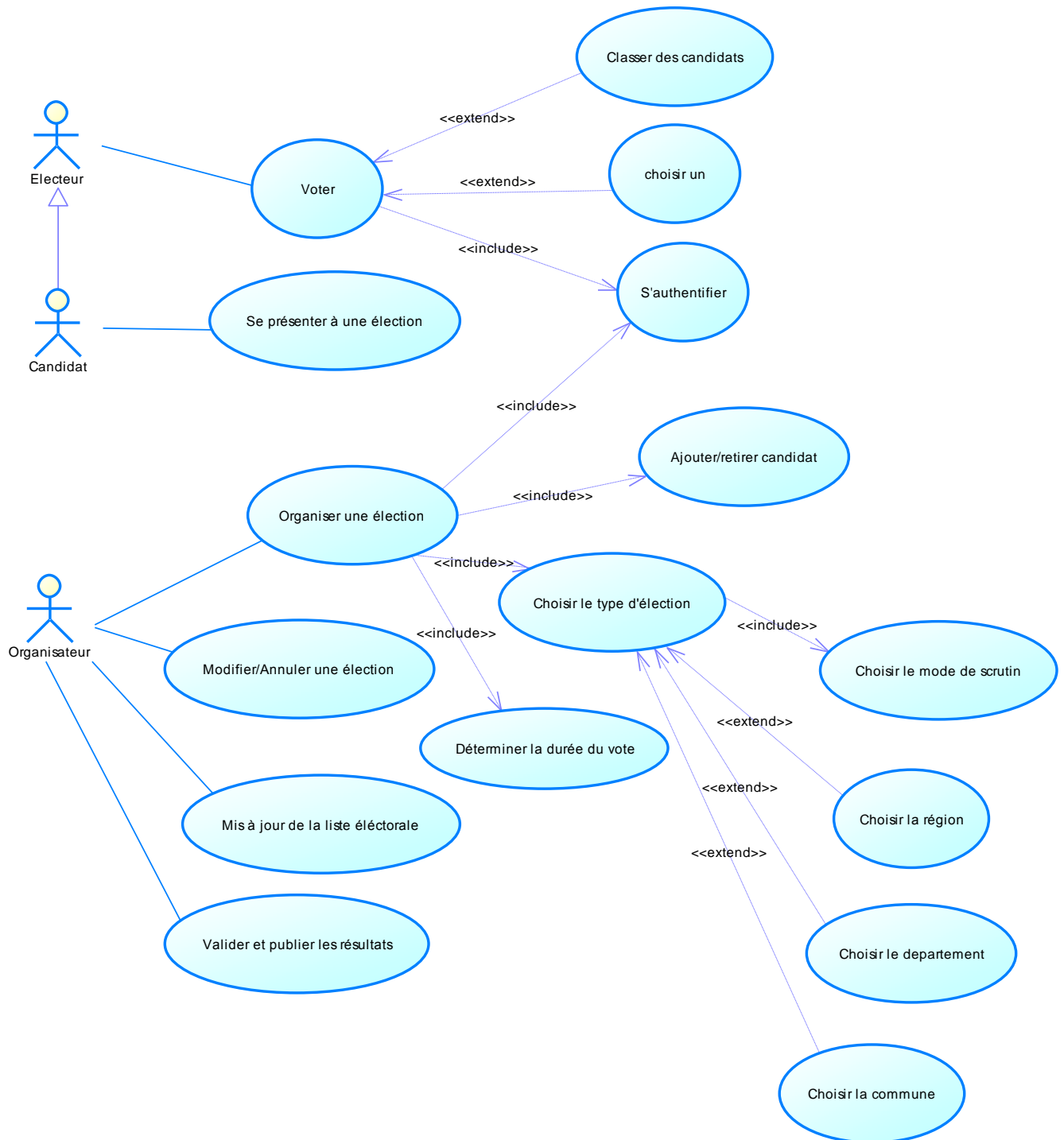
Le système assimilera différents type d'élections selon les divisions territoriales.

Parti :

Un parti est une organisation politique. Le système assimilera ce concept. Chaque candidat à une élection doit appartenir à un et un seul parti. Les candidats indépendants ne sont pas encore pris en compte.

III. Description de l'architecture

Diagramme de Cas d'utilisation



Etant donné qu'on a listé les fonctionnalités dans les spécifications des besoins, il nous paraît adéquat de faire un diagramme de cas d'utilisation afin de donner une vision global du comportement fonctionnel du système eVote

Diagrammes de séquence

Afin de mettre en valeur les scénarios possibles pour les cas d'utilisation ci-dessus, on a décidé de faire les diagrammes de séquence de ces derniers.

Diagramme Ajouter/Retirer un candidat

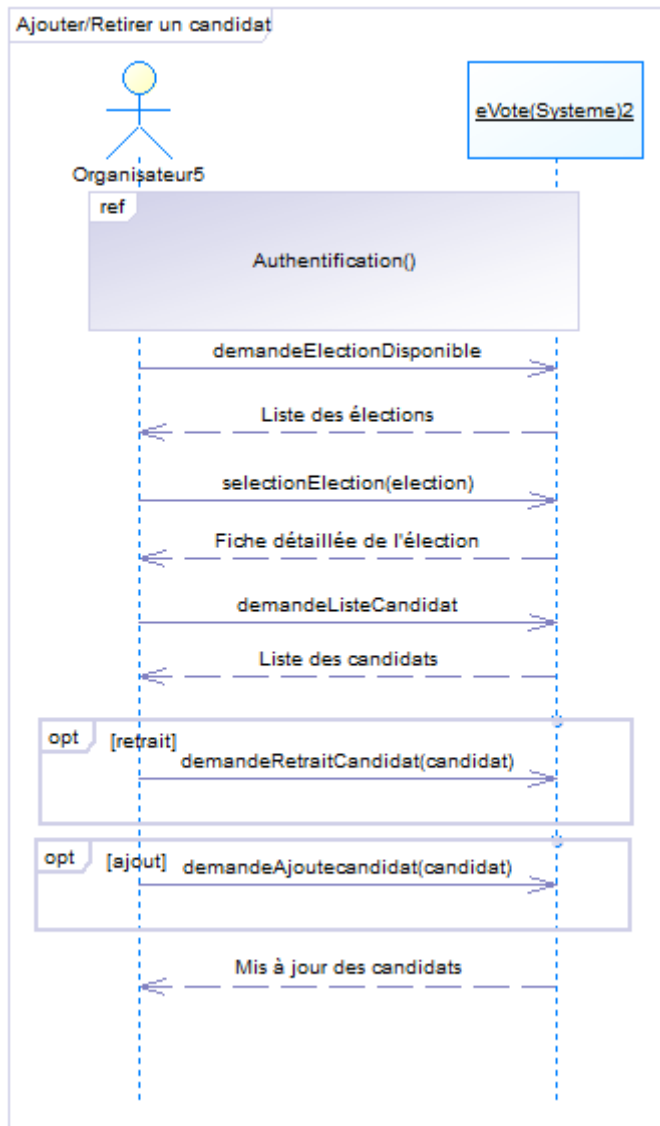


Diagramme Authentication

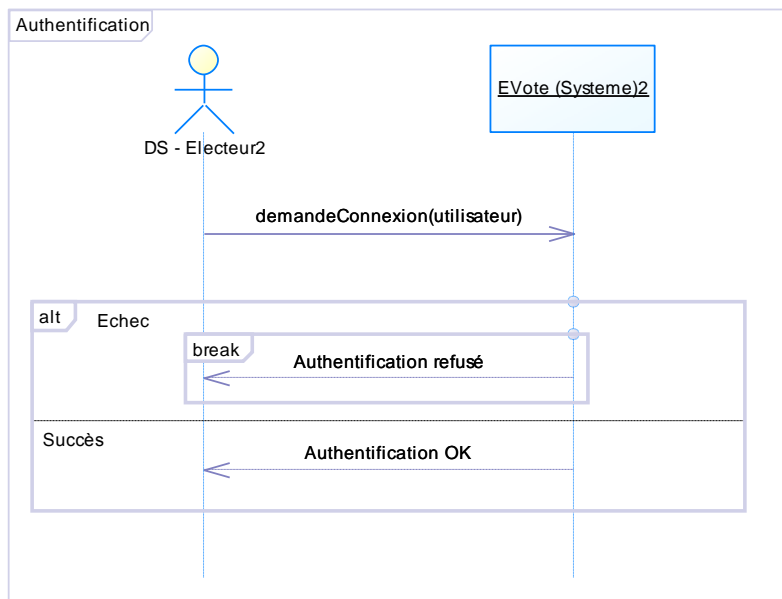


Diagramme Mise à jour de la liste électorale

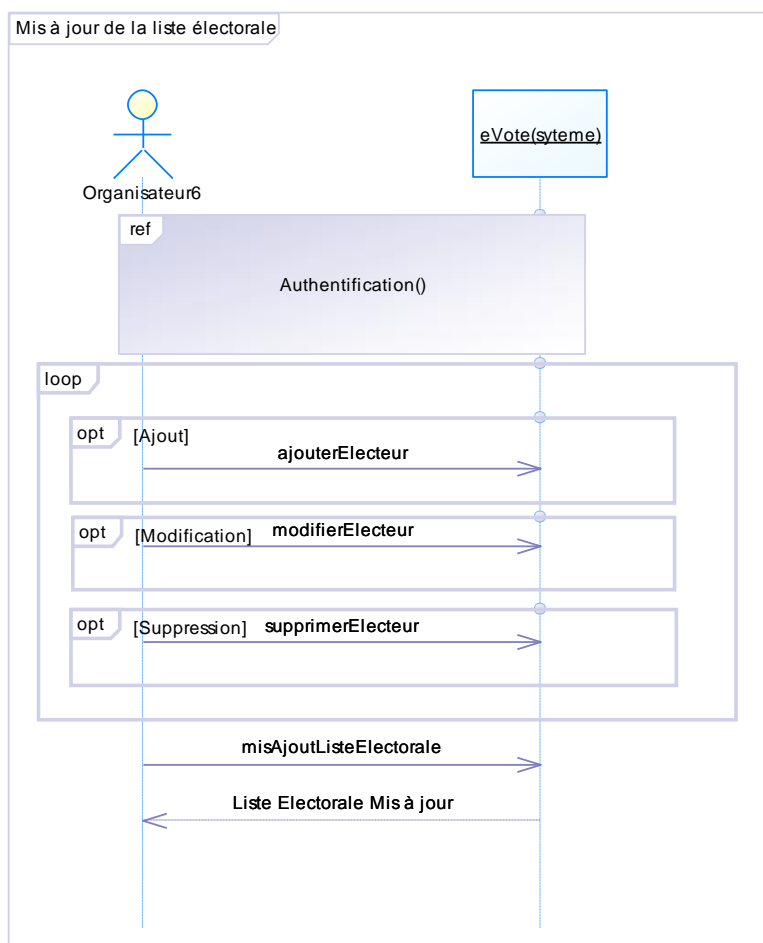


Diagramme Modifier/Annuler une élection

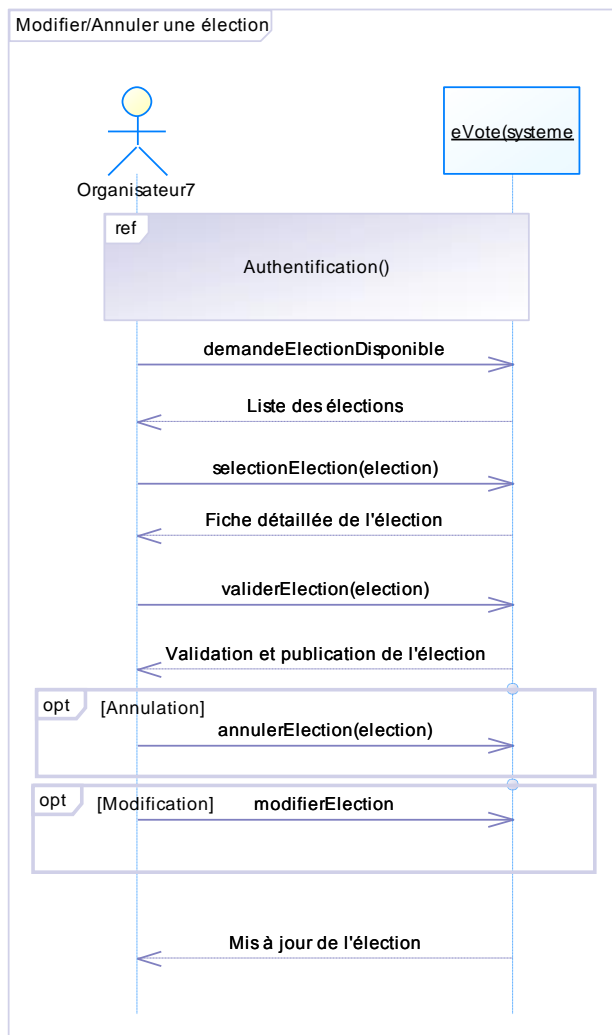


Diagramme Organiser une Election

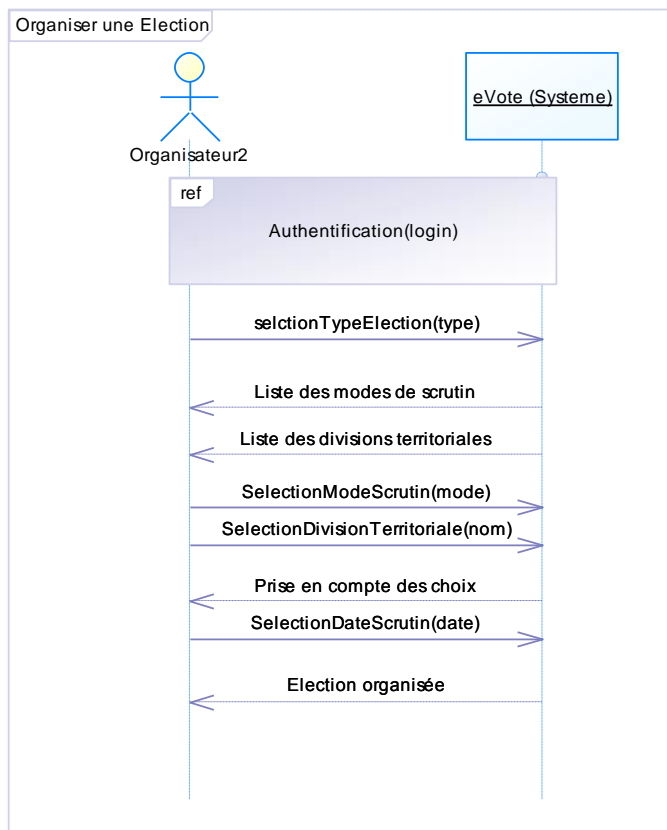


Diagramme Valider Election

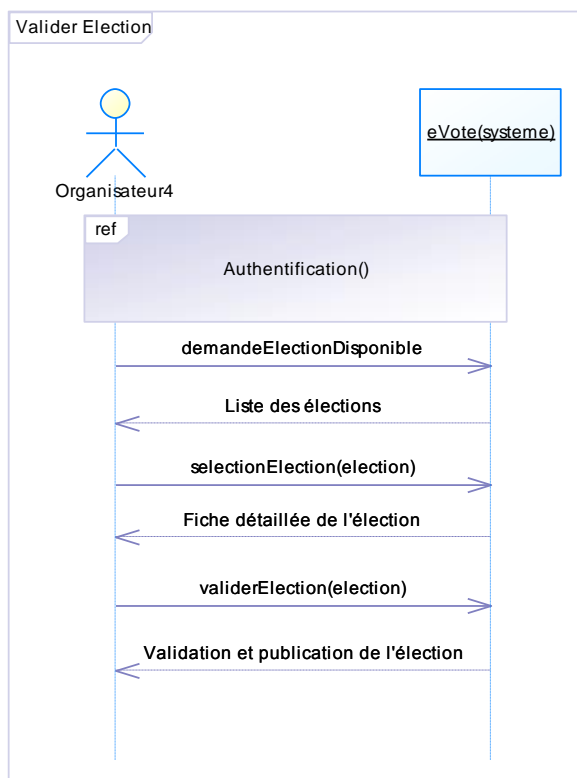
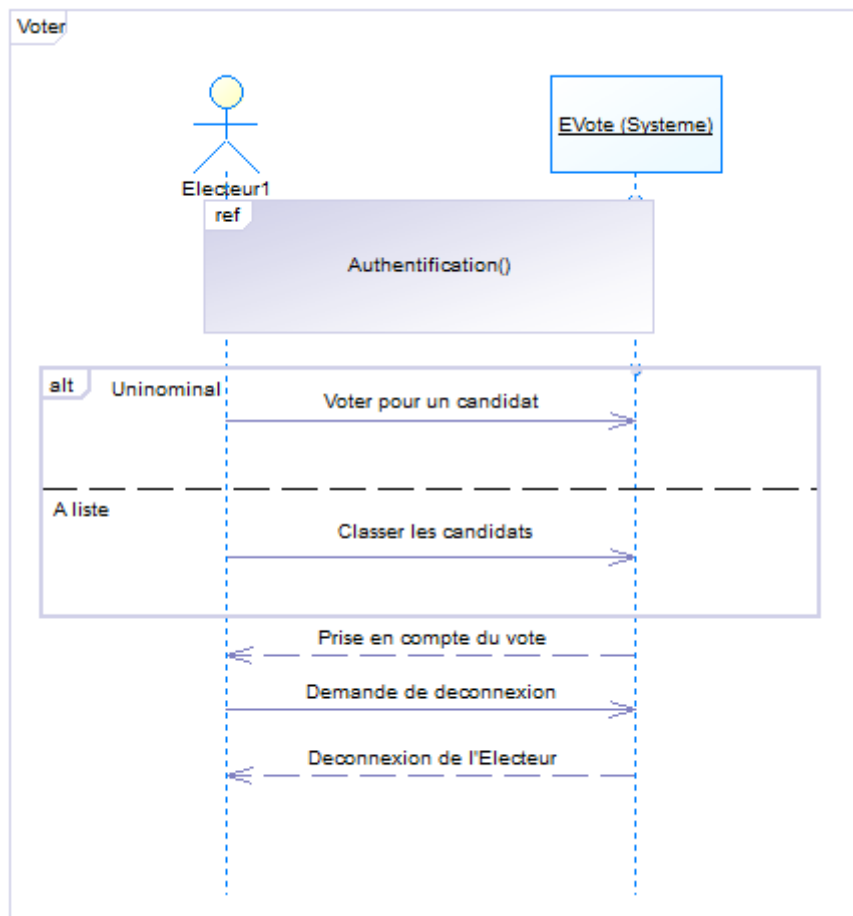
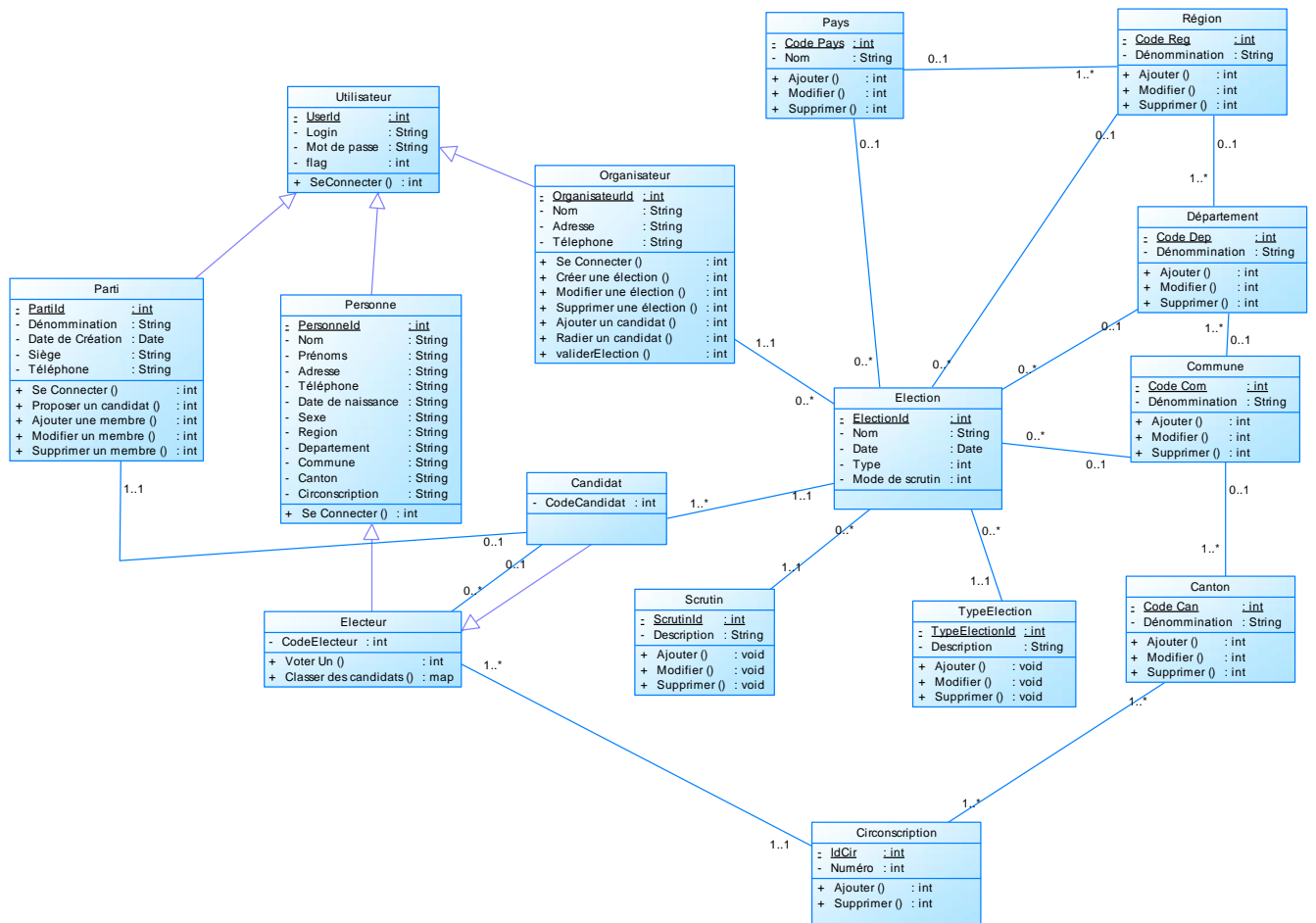


Diagramme Voter



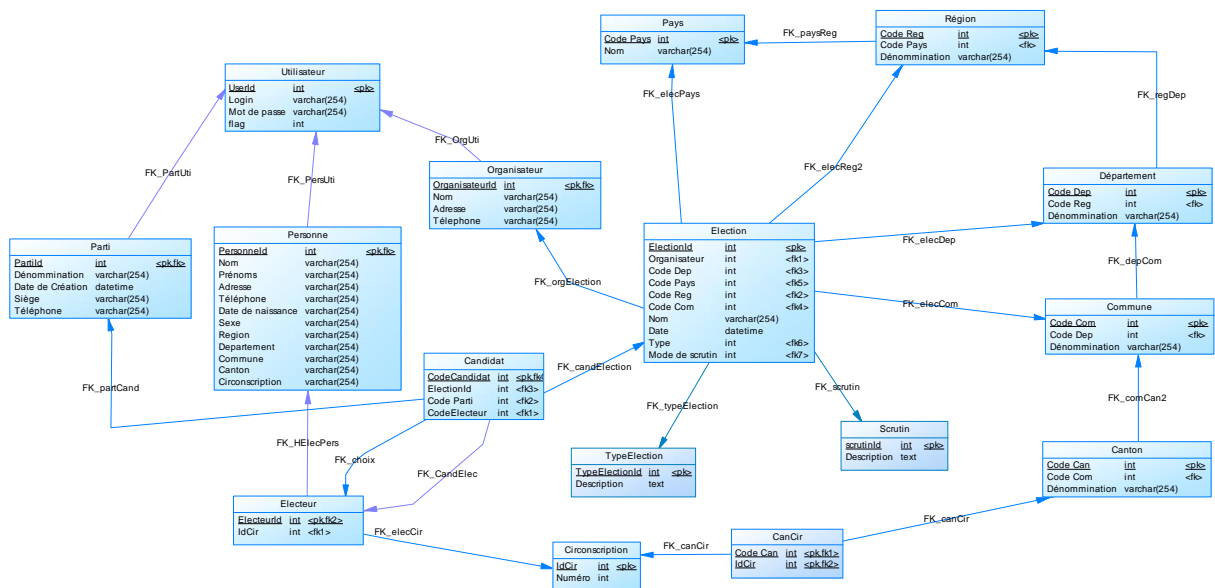
Diagrammes de classes

En vue des implémentations, on a fait le diagramme des classes pour mettre en valeur les classes à implémenter éventuellement et aussi pour faire le futur modèle physique



Modèle physique

Dans ce projet on a décidé d'utiliser une base de données MySQL pour la persistance des données. Du coup on avait besoin d'un modèle physique dérivé du diagramme de classes.



IV. Evolutions envisagées

Candidats indépendants :

Pour intégrer cette évolution, il faut changer la cardinalité entre la table parti et candidat et ajouter les méthodes qu'il faut dans la classe candidat

Gestion des membres d'un parti via le système

On pourra ajouter une relation entre les tables « parti » et « electeur » puis ajouter les méthodes d'ajout, de modification et de suppression dans la classe parti.

Une table de relation entre les tables « parti » et « electeur » sera ainsi créer

Proposition de candidat par les partis via le système

Etant donné que les partis ont maintenant des membres, on pourra ajouter une fonctionnalité de proposition de candidat aux organisateurs d'une élection. La candidature arrivera aux organisateurs qui auront la tâche de valider ou refuser la candidature.

On pourra ajouter des méthodes dans la classe parti sans pour autant modifier la structure de la base parti.

Validation ou refus de candidature

Comme cité ci-dessus, pourra ajouter des fonctionnalités de validation et de refus dans la classe organisateur.

En cas de validation de candidatures, il y aura un appel de la fonctionnalité ajout de candidat actuellement.

Nouveau type d'utilisateur

On pourra ajouter de nouveaux types d'utilisateurs sans problème en faisant un héritage au classe utilisateur.

Par exemple, on pourra ajouter un type observateur. Ces utilisateurs auront pour rôle de l'évolution de l'élection.