

# Rapport de projet

# **eVote**

Cours: **Programmation Objet Concepts Avancés**

Groupe: ***Votalisk***

Members: **Josian Chevalier**  
**Vladislav Fitz**  
**Quỳnh Nga Nguyễn**  
**Thomas Salmon**  
**Mohamed Skhiri**

Professeurs: **M. Yann Régis-Gianas**  
**M. Stefano Zacchiroli**

## 1. Interprétation du sujet

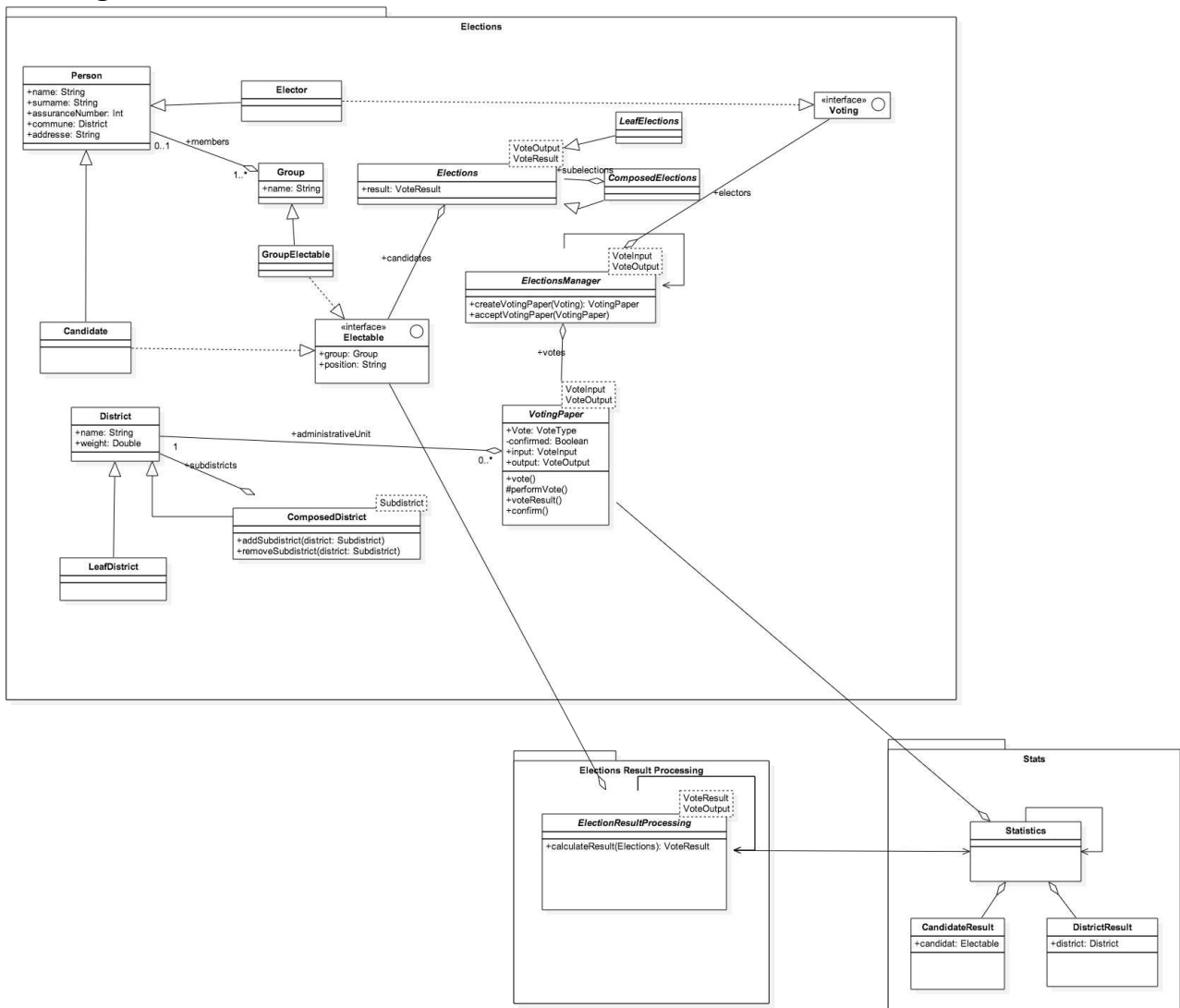
L'objectif de ce framework scala est de fournir un environnement de gestion électorale, il ne s'agit donc que d'un cœur. L'intérêt étant que l'on puisse, à souhait, y ajouter facilement la méthode électorale (ie son algorithme) voulue pour une application.

## 2. Concepts

Votalisk laisse l'application, qui l'utilise, décider de la configuration des élections, aussi on attendra seulement les données et exécuter la stratégie fournie par l'utilisateur.

## 3. Description de l'architecture

### 3.1. Diagramme de Classes



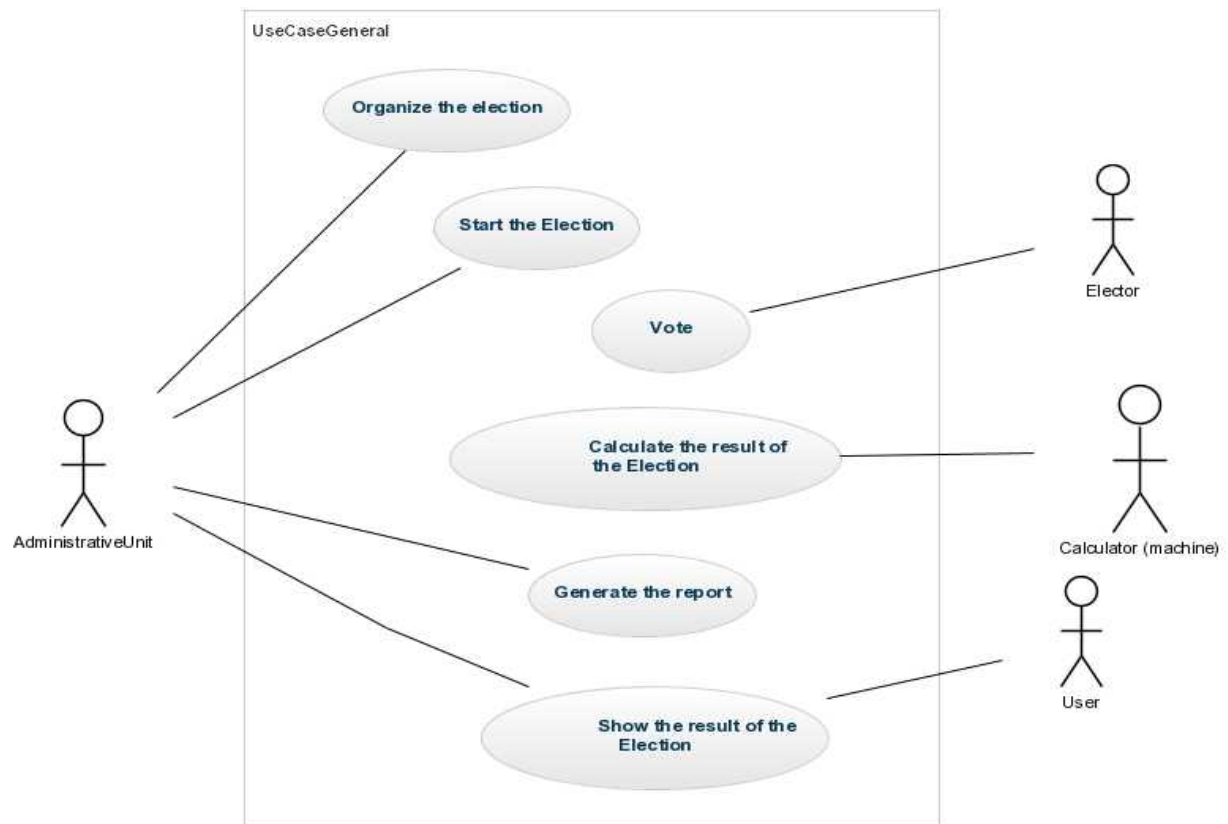
### Description diagramme de classes

- Le lien entre Votalisk et l'application qui l'embarque se fait par la classe **ElectionManager**, avec celle-ci on gère les permissions de votes ainsi que le stockage des scrutins, des candidats, des électeurs...
- La partie **Election** du framework est un **Composite**, ceci afin de gérer les élections à tours successifs.
- Les élections communales, régionales, nationales, etc sont toutes représentées par la partie **Districte**, également **Composite**. On peut donc présenter un territoire électoral ainsi que ses divisions.
- VotingPaper** permet de représenter les scrutins.
- VoteResult**, **VoteInput** et **VoteOutput** sont des types qui sont à définir.
- Le package statistique permet de voir les statistiques sur un candidat (**CandidateResult**) ou sur

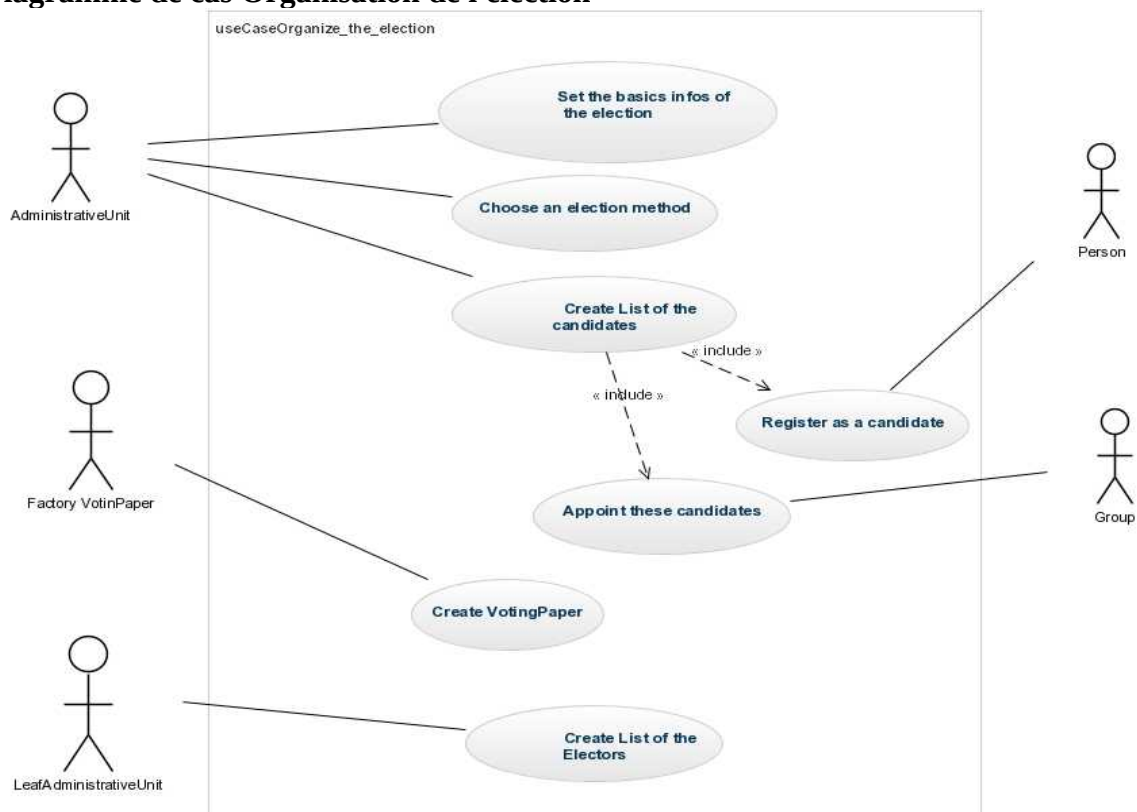
- un district (DistrictResult)
- C'est ce package qui permet de déterminer les résultats dans ElectionResult.

### 3.2. Diagrammes de cas d'utilisation

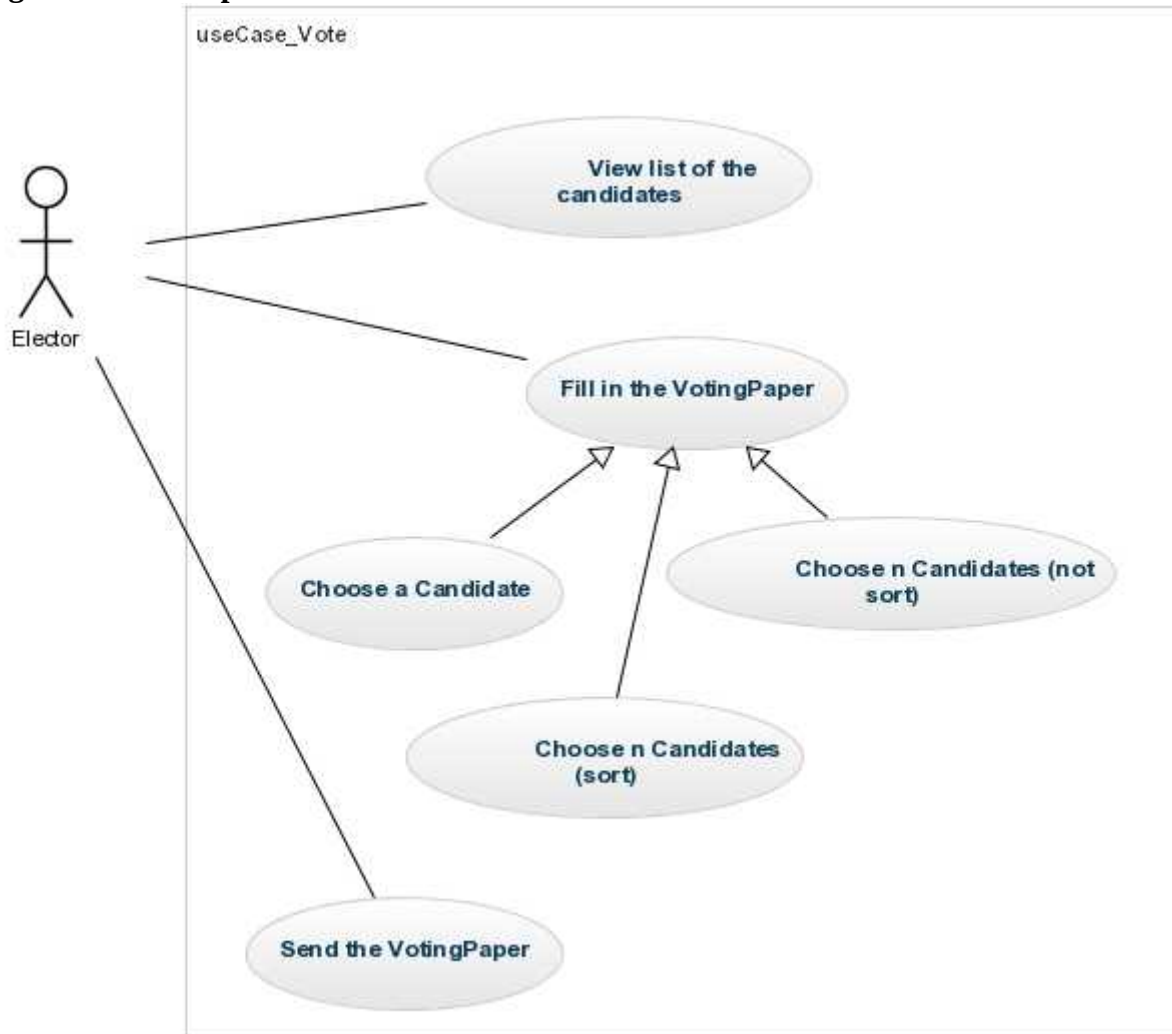
#### \* Diagramme général



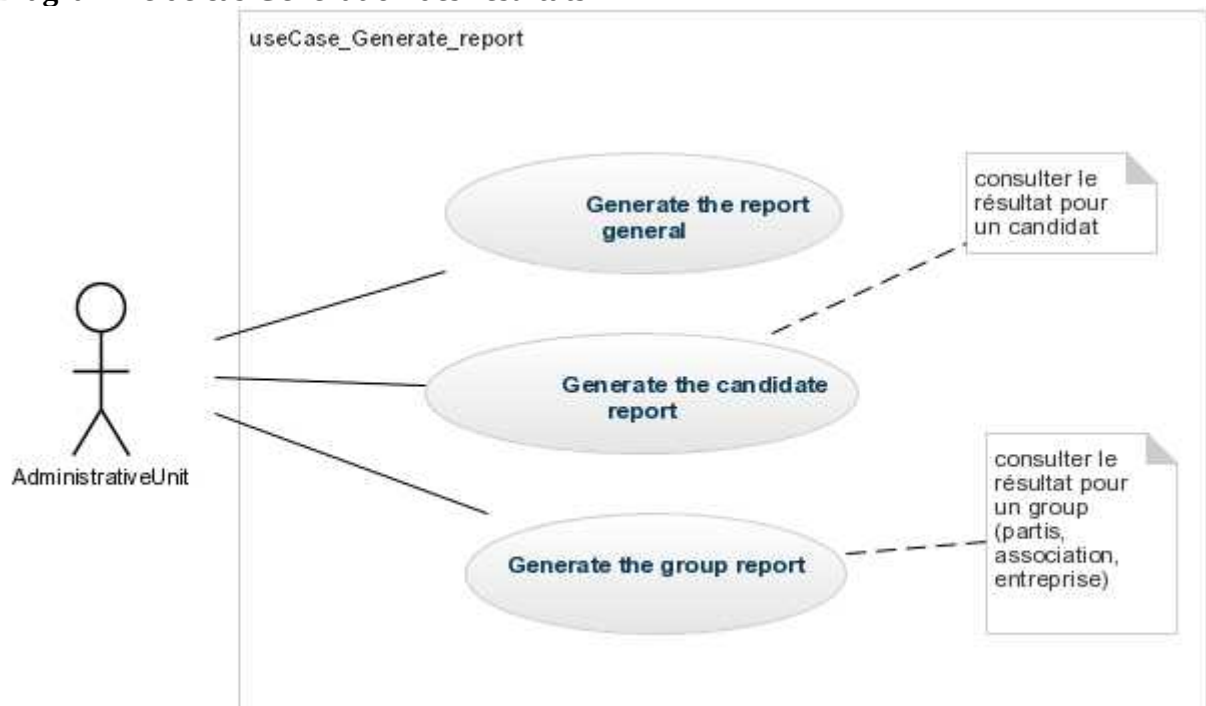
#### \* Diagramme de cas Organisation de l'élection



**\* Diagramme de cas pour le Vote**

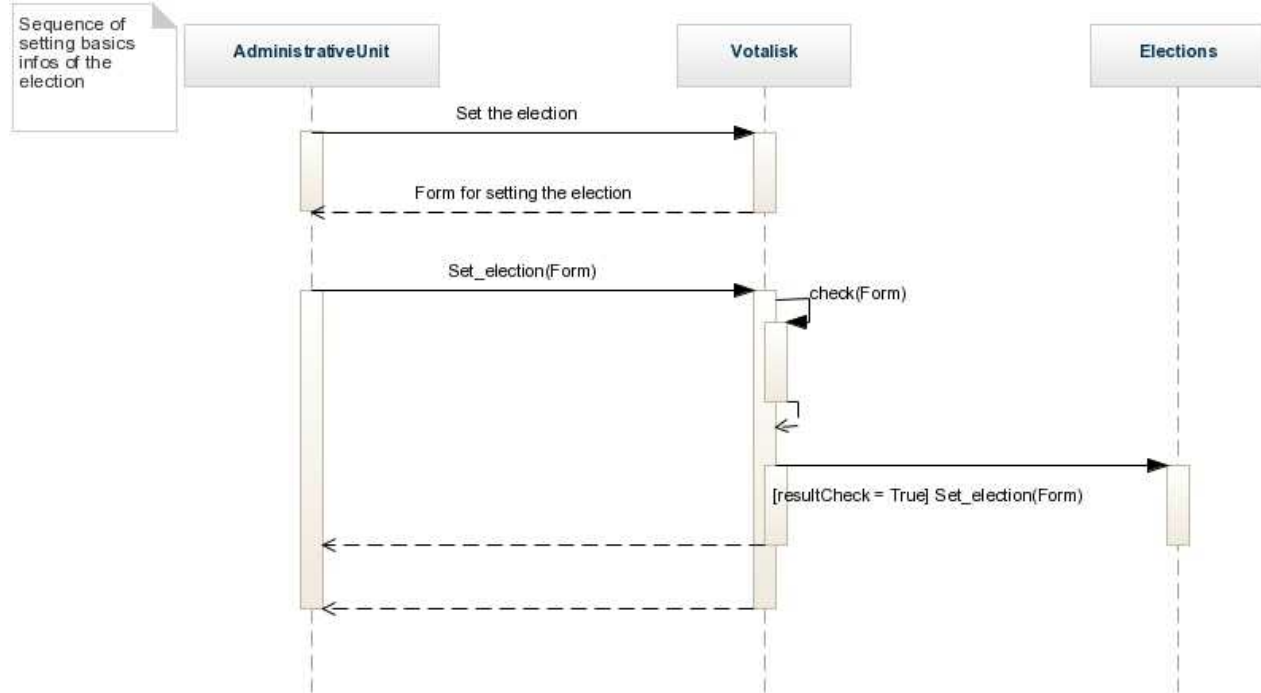


**\* Diagramme de cas Génération des résultats**

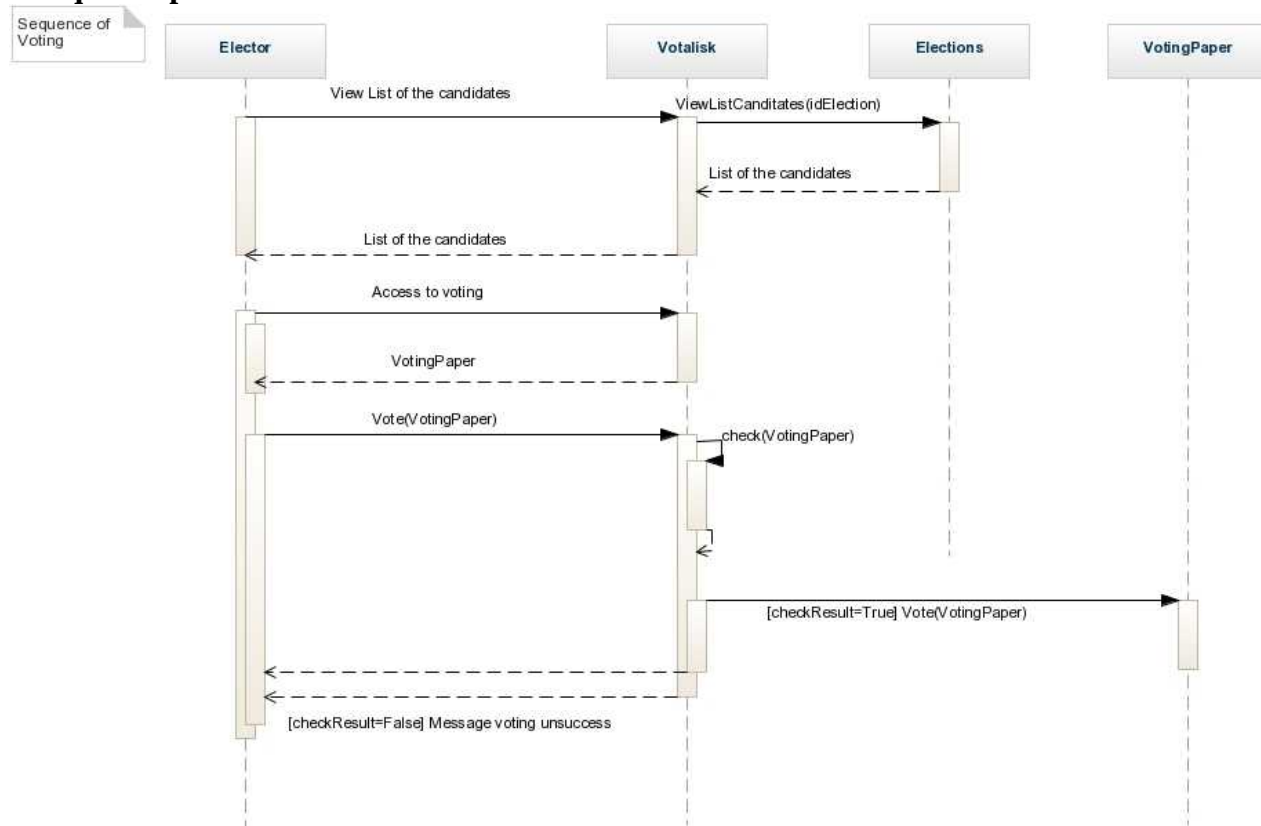


### 3.3. Diagrammes de séquence

#### \* Séquence de mise en place de l'élection



#### \* Séquence pour le Vote



### 4 . Extensions envisagées

- Les méthodes d'élections de différents pays et organisations (France, Union Européenne, etc)
- Nouveaux types de vote (par exemple par préférence, etc): on le fait facilement parce que le

bulletin de vote est une structure non définitive. Pareil pour les différents moyens de traiter les résultats, les résultats associés aux candidats sont des types de données fait pour évoluer de même en ce qui concerne le type de candidats ou d'électeurs (ex : voter pour des mesures, faire voter des organismes).

- Maintenant que on y pense il y a rien qui empêche de bypass l'anonymat dans des elections type vote à l'assemblée ou les votes sont connus.
- Enfin ça permet de gérer n'importe quel type de vote, y compris des trucs par "délégation" vu qu'on a district qui peut représenter n'importe quel groupe de vote, ou autre.