

# RAPPORT

Modélisation de systèmes électoraux et dynamique de vote

## Répartition des rôles

Etant donné la situation de Auxane, la répartition des rôles dans ce projet n'a pas pu être équitable. Auxane vous a déjà communiqué par mail des informations complémentaires sur sa situation.

*Tableau de répartition des rôles :*

Date	Tâche effectuée
15 Décembre	Création d'un premier diagramme UML et de Classes
14 Janvier	Création du git
16 Janvier	Début du code : création classe Électeurs et Candidats
17 Janvier	Création d'une classe Scrutin
26 Janvier	Création d'une interface et remaniement des héritages des classes
28 Janvier	Création d'un scrutin majoritaire à 1 tour (oubli de push sur le git)
30 Janvier	Création d'un scrutin majoritaire à 2 tours
30 Janvier	Remaniement des diagrammes UML et Classes
30 Janvier	Génération de la JavaDoc

*Légende :*

Alice	
Auxane	
Les deux	

## Difficultés rencontrées et solutions

### Création des premières classes :

Pour initialiser et coder les premières classes j'ai mis beaucoup de temps pour comprendre quels seront les attributs qui me seront utiles.

Au tout début, j'avais créé une classe abstraite "Personne" qui était la classe mère de "Electeur" et "Candidat". Après plusieurs tests j'ai décidé de garder uniquement la classe "Electeur" en tant que classe mère de "Candidat".

### Création de la classe Scrutin :

Pour créer cette classe je me suis aidée du travail qu'a effectué Anthony Gonzalez. Il m'a ensuite expliqué comment il avait codé la méthode "preference" pour choisir pour quel candidat un électeur allait voter.

Le code du scrutin majoritaire à 2 tours était plus simple à faire car il ressemble beaucoup à celui à 1 tour.

### Diagrammes :

Quand le code est terminé et que la javadoc a été générée Auxane a pu remanier le diagramme UML et le diagramme de Classe avec ce qui a été effectué.