|  |  |  |
| --- | --- | --- |
| Cas d’utilisation | Acteur | Message émis/reçus par les acteurs |
| Authentification | Electeur | **Emet :** scan du code QR, pour authentification. |
| Système | **Reçu :** Demande d’authentification et informations sur l’électeur. |
| Afficher les candidats correspondants | Système | **Emet**: listes des candidats issues de la même circonscription électorale que l’électeur |
| Electeur | **Reçu :** listes des candidats correspondant |
| Effectuer vote | Electeur | **Emet :** vote un candidat selon la catégorie d’élection |
| Système | **Reçu :** vote affilié à un et un seul candidat |
| Observer résultats | Electeur | **Emet :** demande d’observation des résultats de l’élection en cours |
| Système | **Reçoit :** demande de visualisation des resultats  **Emet : liste des candidats ayant obtenu des voix** |

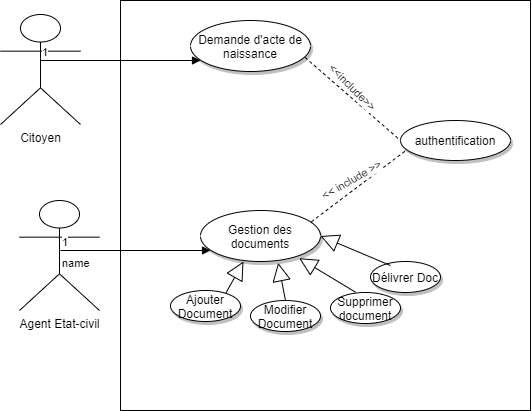
Par la suite nous illustrons graphiquement ce tableau par un diagramme de cas d’utilisation global (voir la figure 4.2 voir ci-dessous)

Figure 1 diagramme de cas d'utilisation

1. **Capture des besoins techniques**

**ENVIRONNEMENT DE REALISATION**

Nous avons eu recours à plusieurs matériels et techniques afin de pouvoir mettre en place cette application :

1. Matériel de base

Le développement de l’application sera réalisé sur un ordinateur portable ayant les caractéristiques suivantes :

|  |
| --- |
| CARACTERISTIQUES MATERIELS |
| Ordinateur : HP laptop 15-bs1xx |
| Ecran : 15 pouces |
| Processeurs : Intel(R) Core(TM) i3-8130U CPU @ 2.20GHz 2.20 GHz |
| Disque dur : 1 To |
| RAM :8,00 Go |

1. CHOIX DE LA PLATEFORME DE DEVELOPPEMENT ET ARCHITECTURE

b.1. Plateforme

Dart est un langage de programmation général, orienté objet, à ramasse-miettes, développé par Google. C'est un langage polyvalent qui peut être utilisé pour créer des applications mobiles, de bureau, de serveur et Web.

Flutter est un kit de développement d'applications mobiles open source développé par Google. Il permet de créer des applications mobiles natives pour iOS et Android à partir d'une seule base de code. Flutter utilise le langage de programmation Dart pour créer des interfaces utilisateur.

1. Firebase

Firebase est une plateforme de développement d’applications mobiles et web, qui propose plusieurs services cloud, tels que l’authentification, les bases de données en temps réel, l’hébergement web, etc.

1. Architecture du système

Le choix de l'architecture d'un système est l'un des aspects les plus importants du processus de développement. L'architecture d'un système définit la structure et la façon dont les différents composants du système interagissent.

Le choix de l'architecture peut avoir un impact significatif sur l'évolutivité, le temps de développement, le coût et les performances du système.

Pour ce qui nous concerne nous avons opté pour une architecture « serverless », cela signifie que l’application ne s’exécute pas sur un serveur physique mais plutôt sur un réseau des serveurs virtuels. Cela permet à l’application d’être plus scalable et résiliente.

Une image contenant texte, diagramme, capture d’écran, conception

Description générée automatiquement

Figure 2architecture sans serveur

**ELABORATION**

Nous découpons notre système en des sous systèmes sous formes de packages en se basant sur le diagramme de cas d’utilisation globale représenté à la figure xx.xx. compte tenu que notre système n’interagit qu’avec un seul acteur ? nous isolons chaque cas d’utilisation tout en proposant un diagramme de séquence et une description textuelle pour chacun d’entre eux.

**Diagramme de cas d’utilisation « Connexion »**

################ »

Figure du diagramme de cas d’utilisation authentification

**DESCRIPTION TEXTUELLE**

|  |  |
| --- | --- |
| Acteurs | Electeur, [[[Serveur ???]]] |
| Cas d’utilisation | 1. Connexion |
| Pré condition | Ouvrir l’application |
| Description du scénario( scénario nominal) | * L’acteur ouvre l’application * Le système donne la possibilité de se connecter via le scan d’un code QR * L’acteur scan le code QR de sa carte d’électeur * Le système vérifie si le numéro associé au QR code est reconnu comme électeur éligible * Le système valide la connexion * La connexion est effectuée |
| Post condition | Connexion et authentification effectuée, le système redirige l’utilisateur vers la page de vote |
| Exception(scénario alternatif) | Code QR inexistant :  Le système affiche un message d’erreur |

La figure suivante représente le diagramme de séquence du cas d’utilisation « Connexion ».

############

Figure du diagramme de séquence « connexion »

Diagramme de cas d’utilisation « Afficher les candidats»

############

Figure du diagramme de cas d’utilisation aff. candidats

**DESCRIPTION TEXTUELLE**

|  |  |
| --- | --- |
| Acteurs | Electeur, [[[Serveur ???]]] |
| Cas d’utilisation | Afficher Candidtats |
| Pré condition | S’être connecté et authentifié |
| Description du scénario( scénario nominal) | * Le système ne renvoie que les candidats correspondants à la même circonscription électorale que l’électeur. * L’électeur observe tous les candidats selon leurs catégories |
| Post condition | L’électeur est en mesure de voter après avoir pris connaissance des candidats de sa circonscription. |
| Exception(scénario alternatif) | -- |

La figure suivante représente le diagramme de séquence du cas d’utilisation « Aff. candidat ».

############

Figure du diagramme de séquence Aff. candidats

* **Le cas d’utilisation « voter »**

############

Figure du diag de CU voter

**Description Textuelle**

|  |  |
| --- | --- |
| Acteurs | Electeur, [[[Serveur ???]]] |
| Cas d’utilisation | voter |
| Pré condition | * S’être connecté et authentifié. * Visualiser les candidats éligibles |
| Description du scénario( scénario nominal) | * L’électeur recherche le candidat de son choix * L’électeur sélectionne et donne sa voix * Le système reçoit les choix de l’électeur et les transmets aux candidats correspondants. |
| Post condition | L’électeur a la possibilité de visionner les résultats de l’élection juste après avoir soumis son vote. |
| Exception(scénario alternatif) | -- |

La figure suivante représente le diagramme de séquence du cas d’utilisation « vote ».

############

Figure du diagramme de séquence vote

* **Cas d’utilisation « Observer les résultats »**

############

Figure du diag de CU observer resultats

**DESCRIPTION TEXTUELLE**

|  |  |
| --- | --- |
| Acteurs | Electeur, [[[Serveur ???]]] |
| Cas d’utilisation | Observer les résultats |
| Pré condition | * S’être connecté et authentifié. * Avoir voté |
| Description du scénario( scénario nominal) | * L’électeur est en mesure d’observer l’évolution des voix en temps réel, du moment où il a déjà effectué son vote |
| Post condition | -- |
| Exception(scénario alternatif) | -- |

La figure suivante représente le diagramme de séquence du cas d’utilisation «observer les résultats ».

############

Figure du diagramme de séquence observer les résultats