

Quantification de l'activité motrice des enfants atteints de TDAH

Présenté par :

Imen Toujeni, Nour El Houda Ben Cheikh, Coline Auber



Sommaire

Introduction

Analyse et spécification du besoin

Choix d'environnement technique et développements

Méthodologie choisie

Réalisations

Difficultés rencontrées

Conclusion et perspectives

Démonstration



Introduction



TDAH

Symptômes

- un déficit de l'attention
- une hyperactivité motrice
- une impulsivité

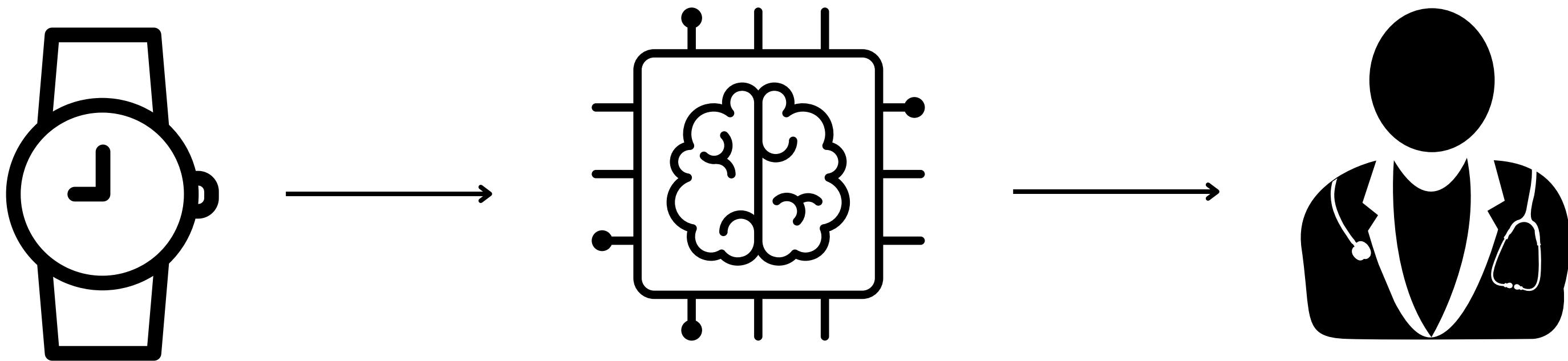
Touche 6% des enfants

Arrive pendant l'enfance

Médecin suit état patient

Généralement avec l'aide des parents

Les Objectifs



Analyse et spécification du besoin



Les Besoins Fonctionnels

Gestion des utilisateurs et authentification

- Admin
- Médecin
- Enfant

Collecte et transmission des données

Visualisation et analyse des données

Les Besoins Non- Fonctionnels

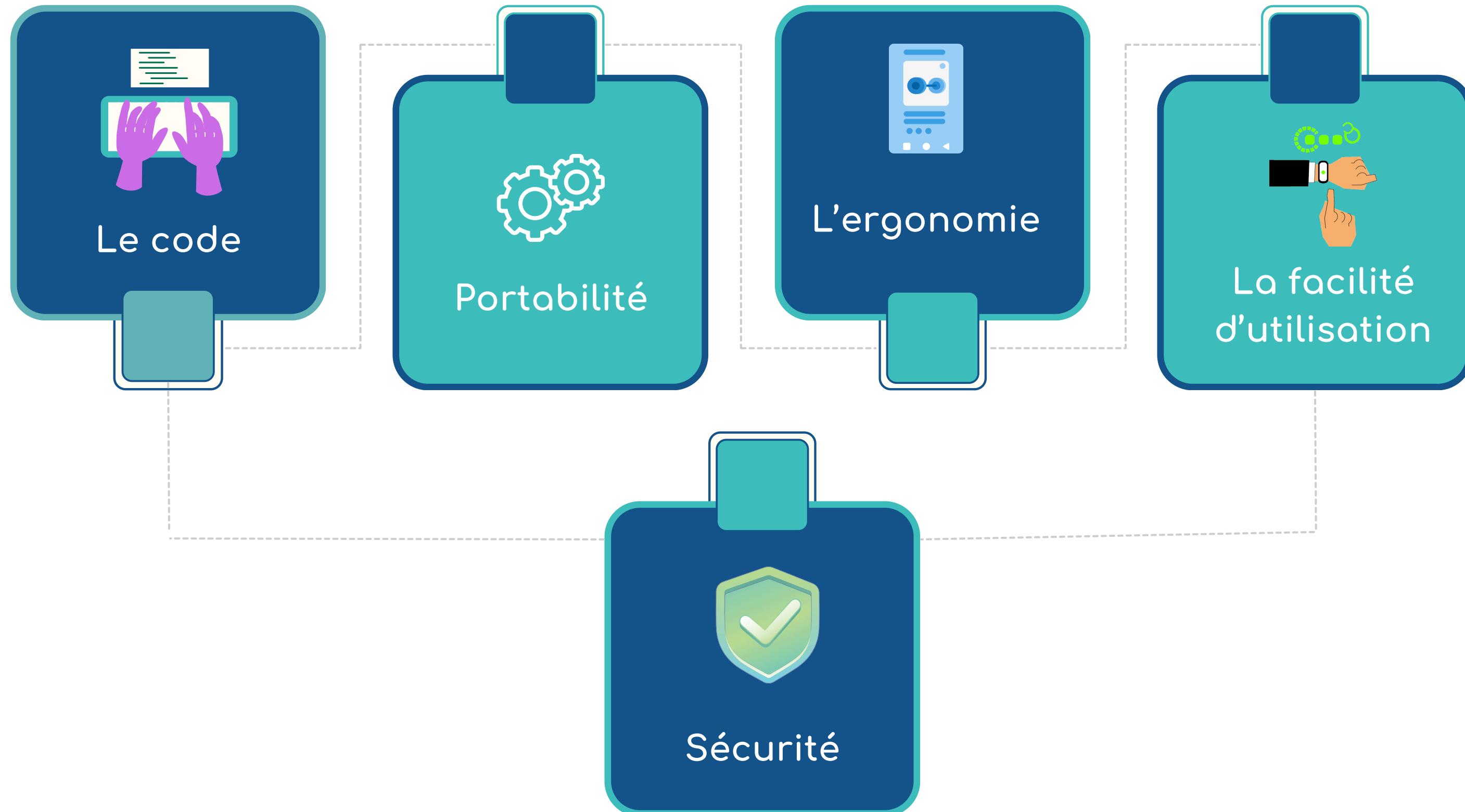
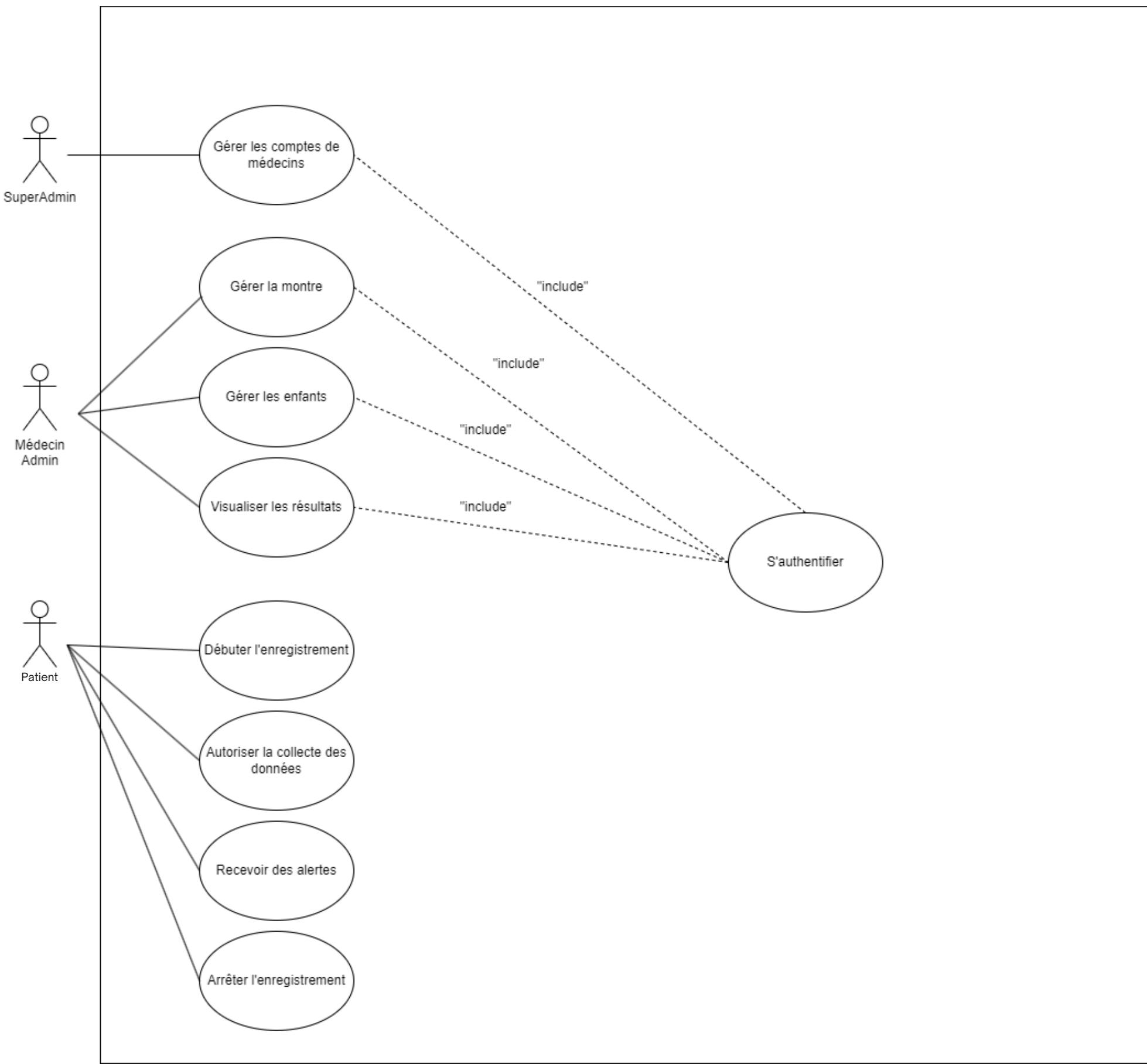


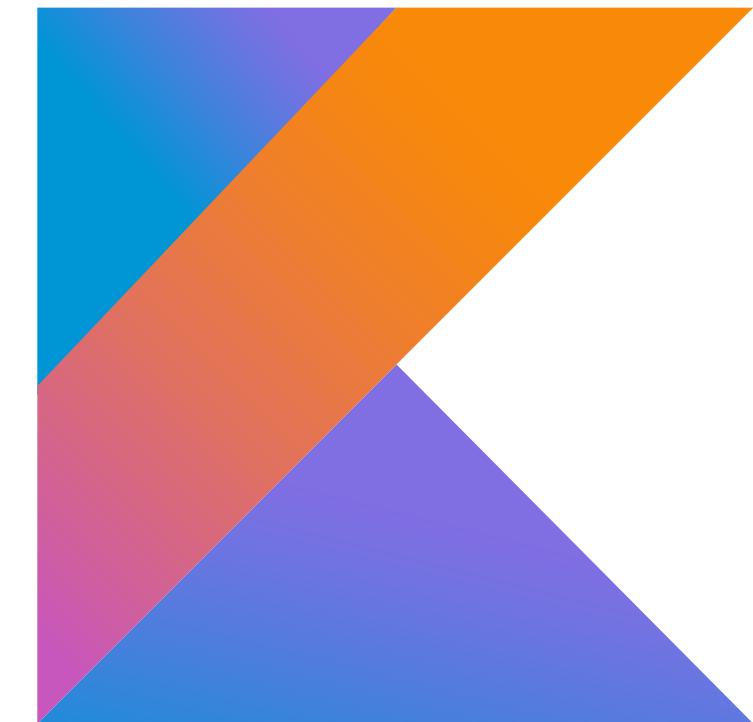
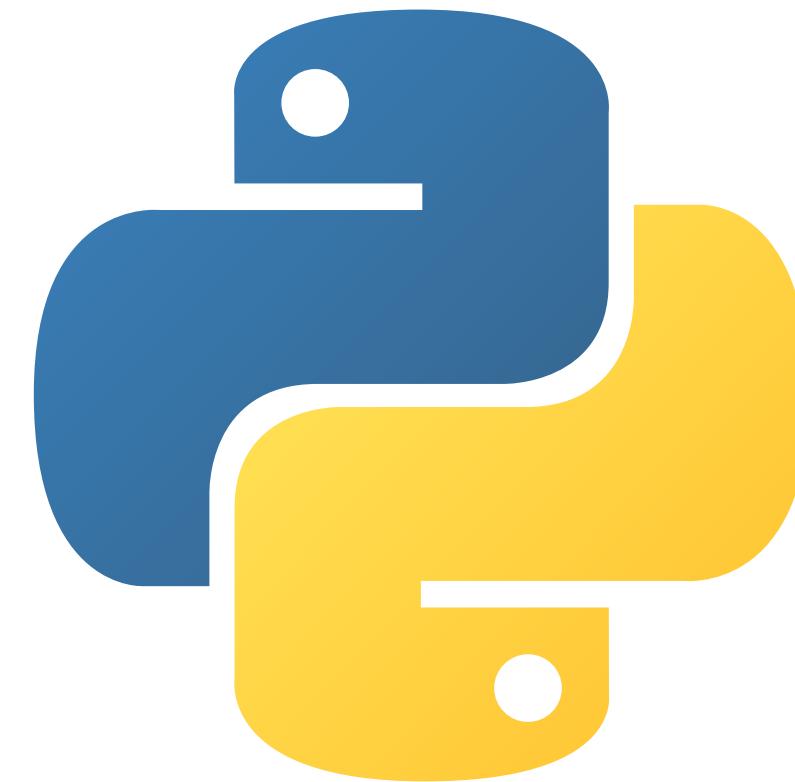
Diagramme de cas d'utilisation global



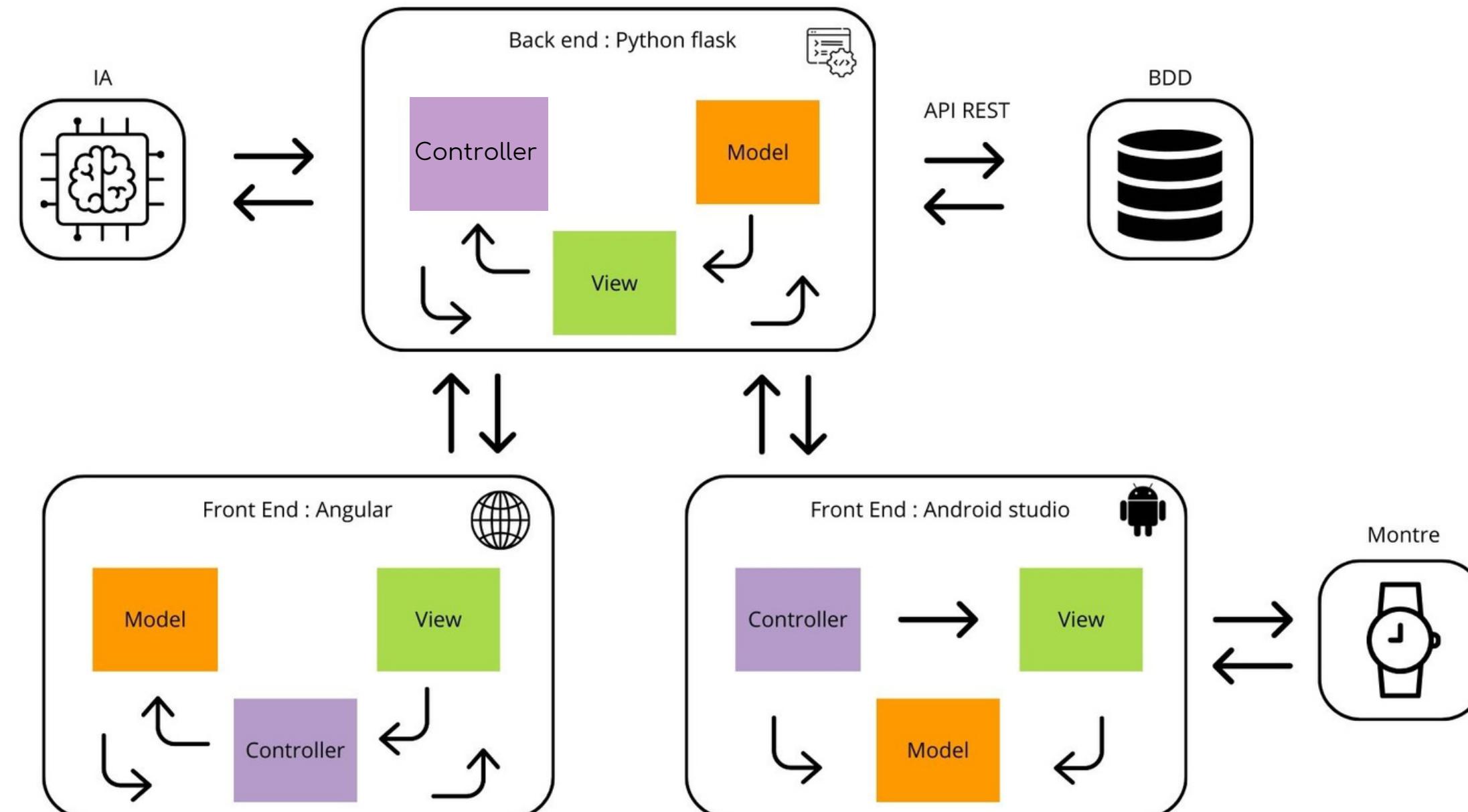
Choix d'environnement technique et développements



Choix d'environnement technique et développements



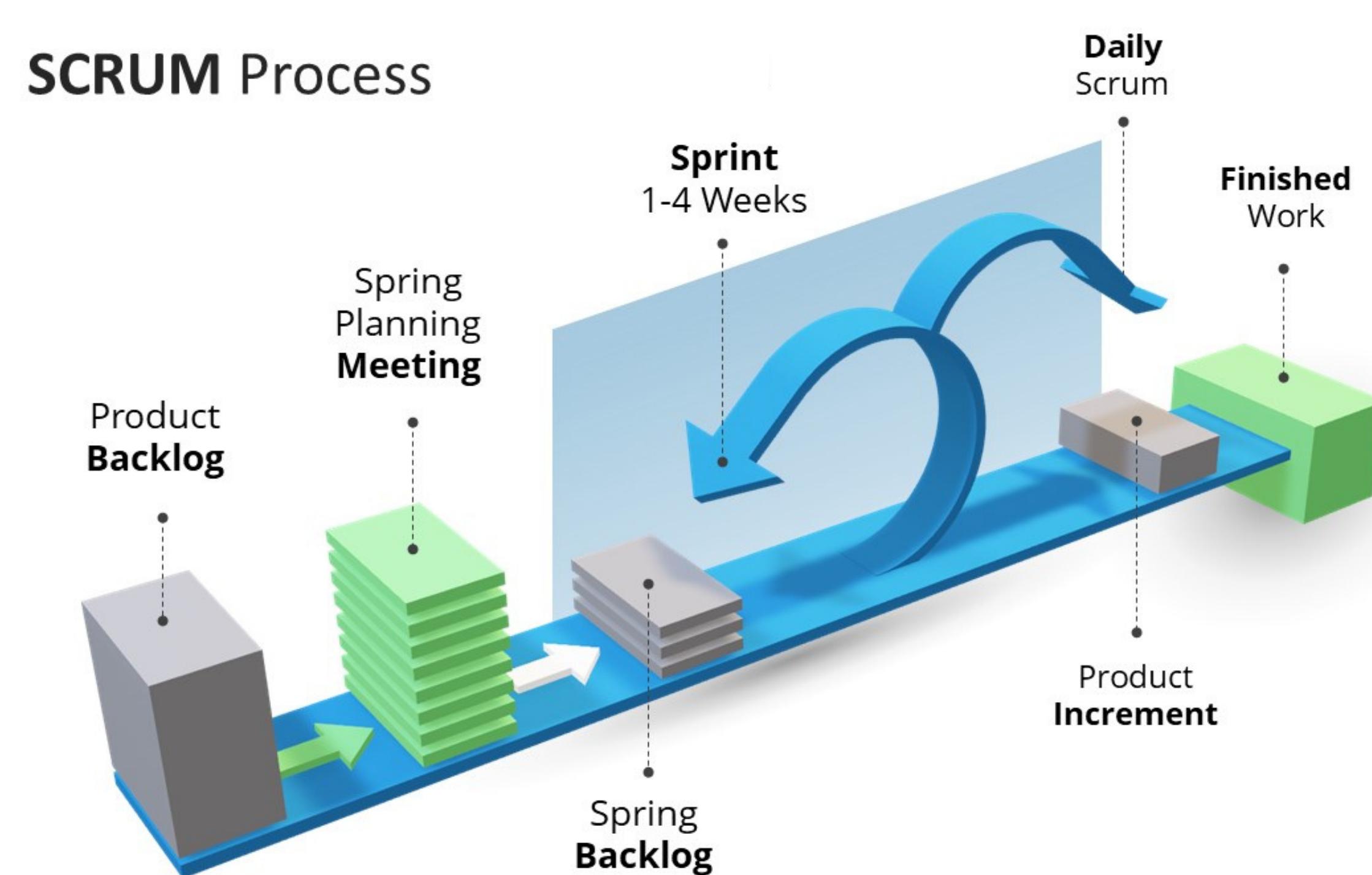
Choix d'environnement technique et développements



Méthodologie choisie



Processus Scrum



Organisation du backlog de produit

Sprint1 :
Gestion des comptes et authentification

**Gestion du compte Médecin
(par SuperAdmin)**

Gestion du compte Enfant (par Médecin)

Authentification (tous les utilisateurs)

Détection de non-port de la montre

Organisation du backlog de produit

Sprint 1 : Gestion des comptes et authentification	Sprint 2: Configuration de la montre
Gestion du compte Médecin (par SuperAdmin)	Saisie de la fréquence de la récolte des données
Gestion du compte Enfant (par Médecin)	Choix des capteurs
Authentification (tous les utilisateurs)	Reçoiit des paramètres
Détection de non-port de la montre	Stockage des paramètres dans la montre
	Visualisation des états de la montre

Organisation du backlog de produit

Sprint 1 : Gestion des comptes et authentification	Sprint 2: Configuration de la montre	Sprint 3 : Collecte des traitements de données
Gestion du compte Médecin (par SuperAdmin)	Saisie de la fréquence de la récolte des données	Autorisation de la collecte
Gestion du compte Enfant (par Médecin)	Choix des capteurs	Début de l'enregistrement
Authentification (tous les utilisateurs)	Reçoiit des paramètres	Envoi et réception des notifications (alertes)
Détection de non-port de la montre	Stockage des paramètres dans la montre	Arrêt de l'enregistrement
	Visualisation des états de la montre	Envoi automatique et manuel des données vers le serveur
		Suppression des données envoyées

Organisation du backlog de produit

Sprint 1 : Gestion des comptes et authentification	Sprint 2: Configuration de la montre	Sprint 3 : Collecte des traitements de données	Sprint 4 : Visualisation des données
Gestion du compte Médecin (par SuperAdmin)	Saisie de la fréquence de la récolte des données	Autorisation de la collecte	Visualisation des données de capteurs
Gestion du compte Enfant (par Médecin)	Choix des capteurs	Début de l'enregistrement	Visualisation des statistiques journalières
Authentification (tous les utilisateurs)	Reçoiit des paramètres	Envoi et réception des notifications (alertes)	Exportation des données en CSV
Détection de non-port de la montre	Stockage des paramètres dans la montre	Arrêt de l'enregistrement	
	Visualisation des états de la montre	Envoi automatique et manuel des données vers le serveur	
		Suppression des données envoyées	

L'équipe de développement



Imen

Développeuse Front-
End Web



Nour

Développeuse Back-End



Coline

Développeuse Front-End
Mobile

Réalisations



Diagramme de classe global

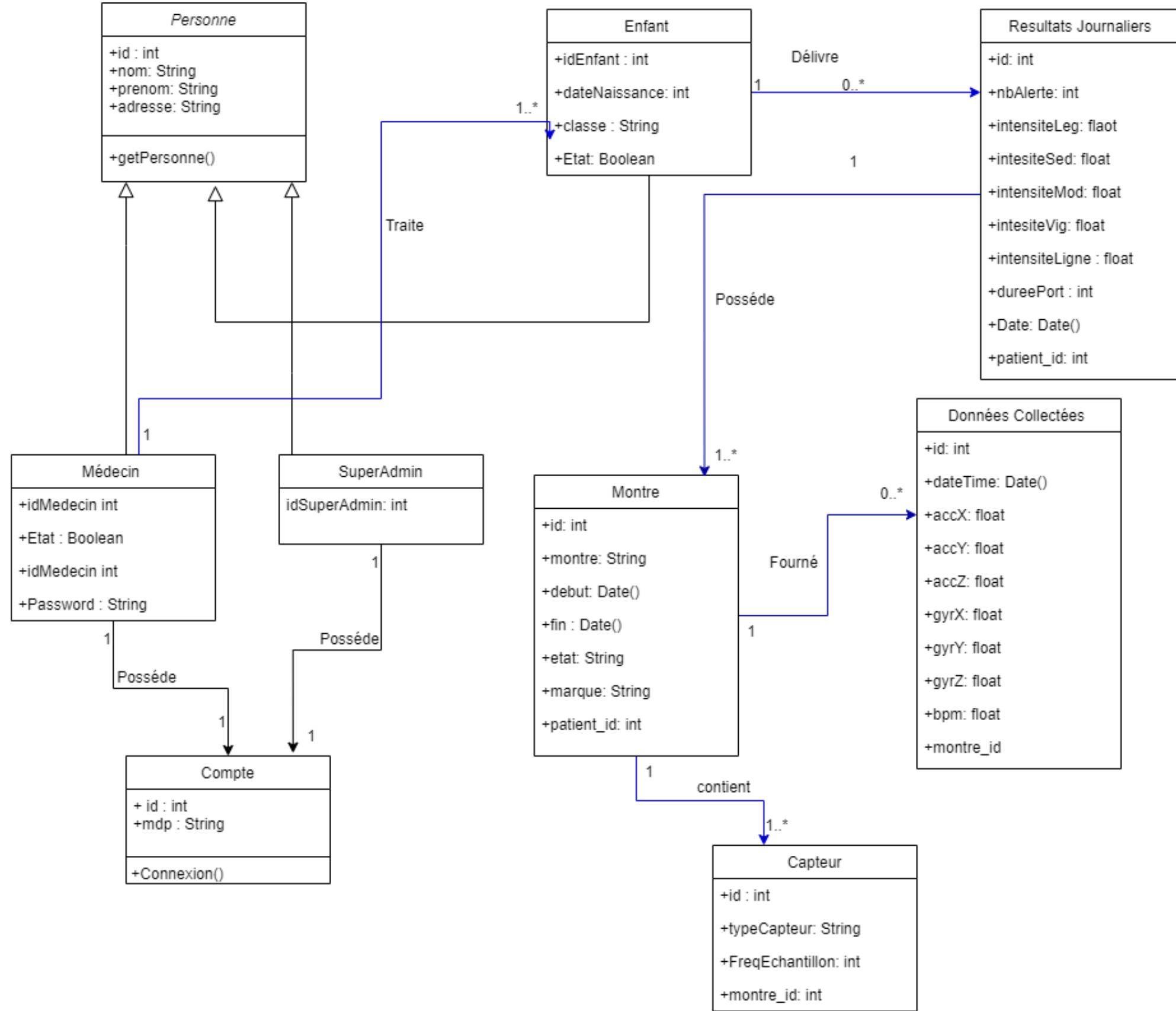


Diagramme de séquence : Collecte et traitements des données

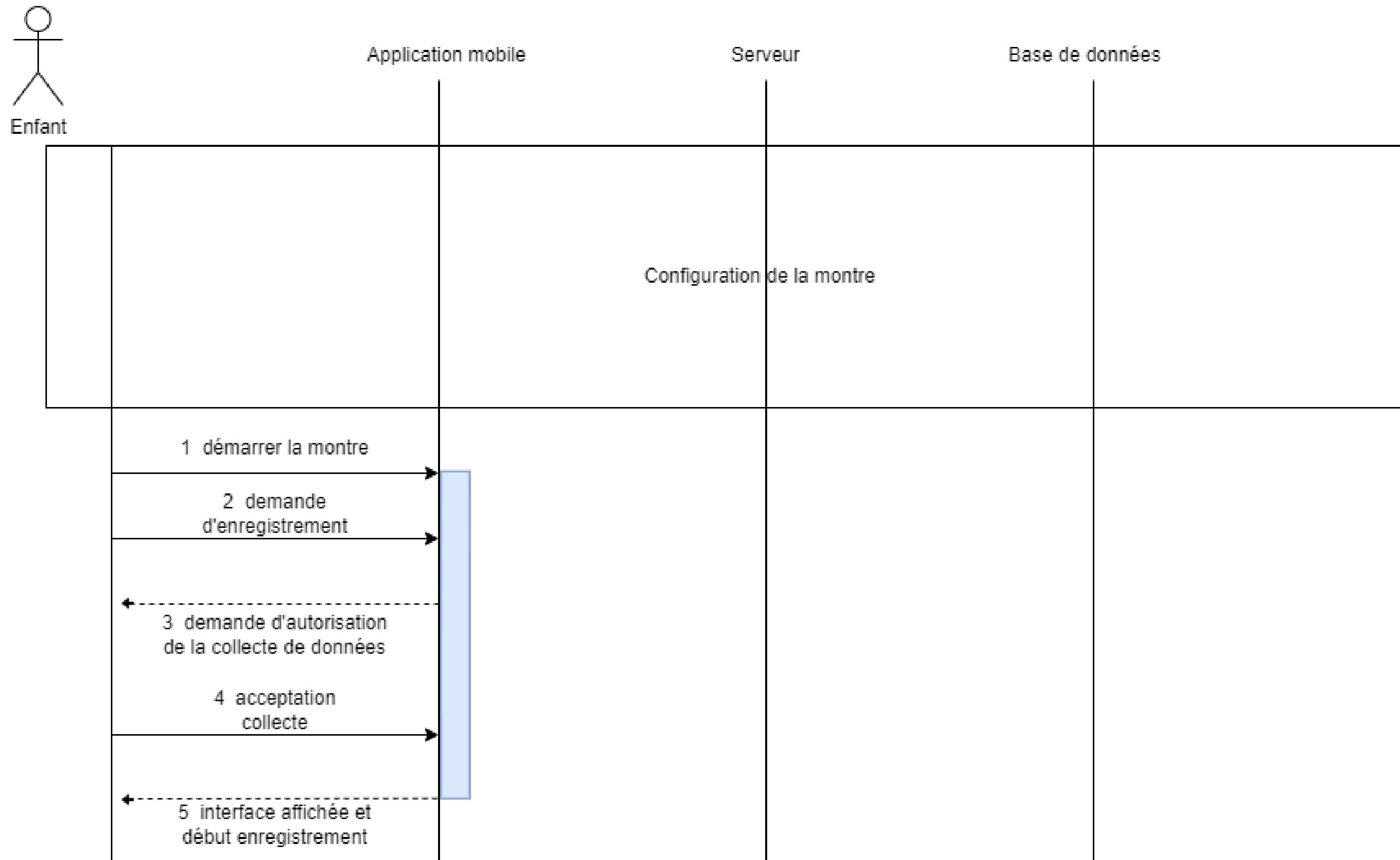
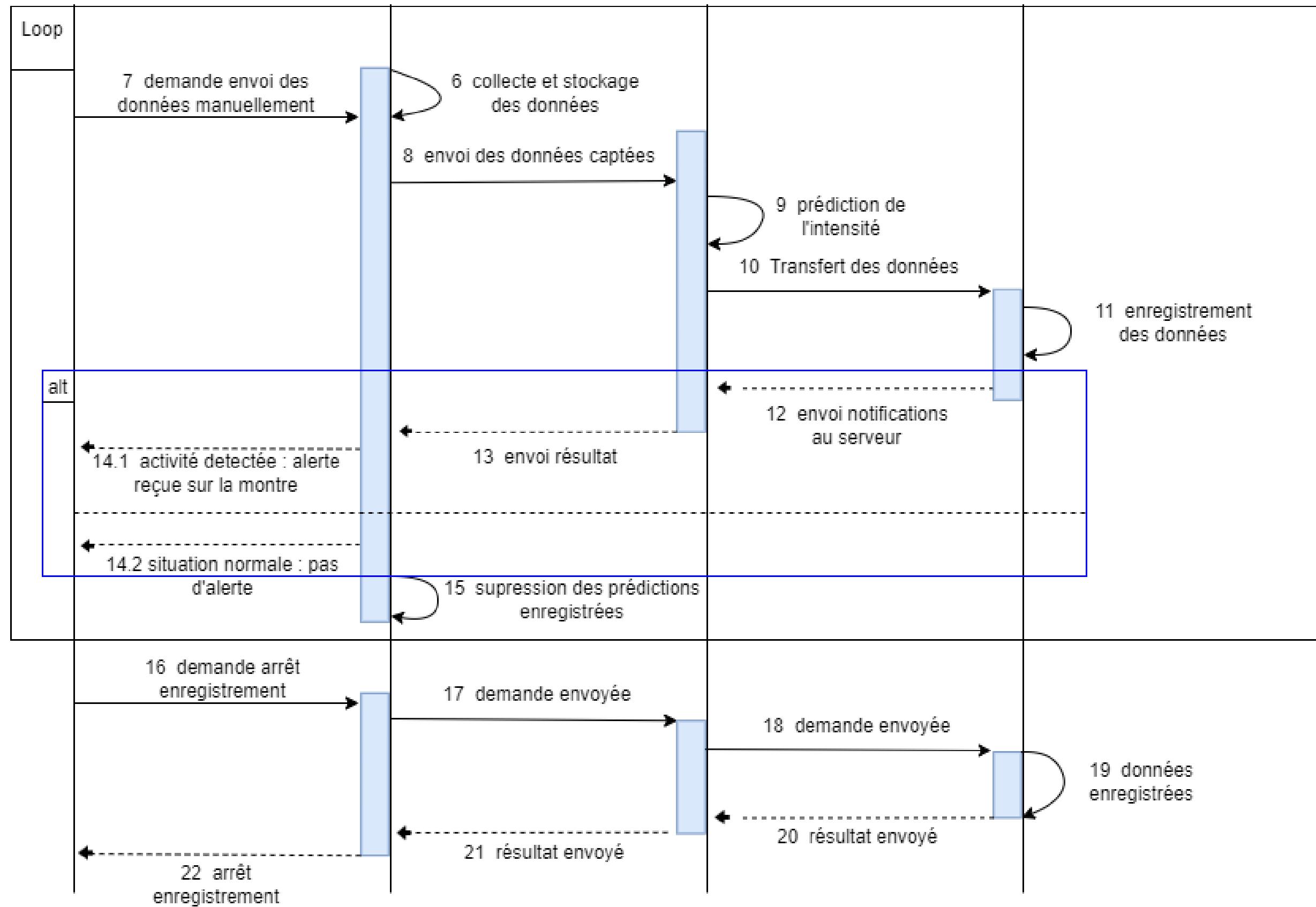


Diagramme de séquence : Collecte et traitements des données



Back End

TDAH API

Overview

ENDPOINTS

- Tokens >
- Users >
- Patients >
- Montres >
- Capteurs >
- Donnee_Collectees >
- Resultat_Journaliers >

SCHEMAS

- User
- UpdateUserUpdate
- StringPagination
- PaginatedUser
- Token
- Empty
- Montre
- PaginatedMontre
- Patient

TDAH API

v1.0

API Base URL

Live Server: <http://127.0.0.1:5000/>

Export ▾

EndPoints

Tokens

- ➊ Create new access and refresh tokens [POST](#)
- ➋ Refresh an access token [PUT](#)
- ➌ Revoke an access token [DELETE](#)
- ➍ Request a password reset token [POST](#)
- ➎ Reset a user password [PUT](#)
- ➏ Initiate OAuth2 authentication with a ... [GET](#)
- ➐ Create new access and refresh token... [POST](#)

Patients

- ➊ Récupérer tous les patients [GET](#)
- ➋ Créer un nouveau patient [POST](#)
- ➌ Récupérer un patient par son ID [GET](#)
- ➍ Mettre à jour un patient [PUT](#)
- ➎ Supprimer un patient [DELETE](#)

Users

- ➊ Retrieve the authenticated user [GET](#)
- ➋ Edit user information [PUT](#)
- ➌ Retrieve all users [GET](#)
- ➍ Register a new user [POST](#)
- ➎ Retrieve a user by id [GET](#)
- ➏ Supprimer un User [DELETE](#)
- ➐ Retrieve a user by username [GET](#)

EndPoints

Montres

- ➊ Récupérer tous les montres GET
- ➋ Créer un nouvelle montre POST
- ➌ Récupérer une montre par son ID GET
- ➍ Mettre à jour une montre PUT
- ➎ Supprimer une montre DELETE
- ➏ montres_checkMontre GET

Resultat_Journaliers

- ➊ Récupérer tous les donnees_collectees GET
- ➋ Créer des donnees_collectees POST
- ➌ Récupérer une resultat_journalier par ... GET

Donnee_Collectees

- ➊ Récupérer tous les donnees_collectees GET
- ➋ Créer des donnees_collectees POST
- ➌ Récupérer une donnee_collectee par ... GET
- ➍ Supprimer une donnee_collectee DELETE

Capteurs

- ➊ Récupérer tous les capteurs GET
- ➋ Créer un nouveau capteur POST
- ➌ Récupérer un capteur par son ID GET
- ➍ Mettre à jour un capteur PUT
- ➎ Supprimer un capteur DELETE

EndPoints

TDAH API

Overview

ENDPOINTS

Tokens

- >Create new access and refresh tokens [POST](#)
- Refresh an access token [PUT](#)
- Revoke an access token [DELETE](#)
- Request a password reset token [POST](#)
- Reset a user password [PUT](#)
- Initiate OAuth2 authentication with a ... [GET](#)
- Create new access and refresh token... [POST](#)

Users

- Retrieve the authenticated user [GET](#)
- Edit user information [PUT](#)
- Retrieve all users [GET](#)
- Register a new user [POST](#)
- Retrieve a user by id [GET](#)
- Supprimer un User [DELETE](#)
- Retrieve a user by username [GET](#)

Patients

- Récupérer tous les patients [GET](#)

Récupérer tous les patients

GET <http://127.0.0.1:5000/api/patients>

Request

Security: Bearer Auth

Provide your bearer token in the Authorization header when making requests to protected resources.

Example: `Authorization: Bearer 123`

Query Parameters

after string write-only

limit integer

offset integer

Responses

200

Auth

Token : eyJhbGciOiJIUzI1NilsInR5cCI6IkpXVCJ9eyJ0b2tlbi

Parameters

after : string

limit : integer

offset : integer

Send API Request

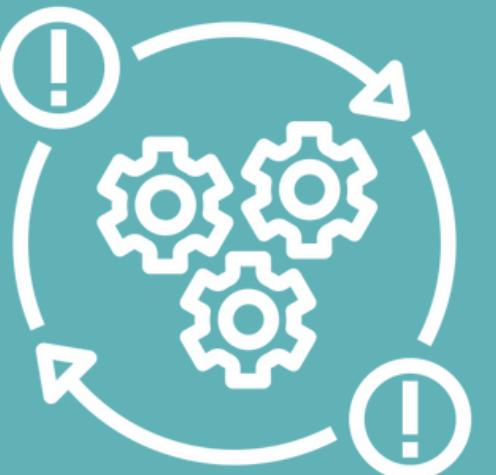
Response

Preview

200 OK

```
{  
  "data": [  
    {  
      "classe": "fie4",  
      "dateNaissance": "16-07-2000",  
      "etat": "actif",  
      "id": 1,  
      "nom": "Dupont",  
      "prenom": "Jean"  
    }  
  ]  
}
```

Difficultés rencontrées



Back-End

- Python Flask

Front-End Web

- Sécurité (token)

Front-End Android

- Tâche de fond (worker)

Conclusion et perspectives



Démonstration



Merci de votre
attention!

Des questions?

