

جامعة ابن زهر
+080.1141 401 4800
UNIVERSITÉ IBN ZOHR



Université Ibn Zohr
FACULTÉ DES SCIENCES
CENTRE D'EXCELLENCE IT - AGADIR

Master d'Excellence

Analytique des Données et Intelligence Artificielle &
Ingénierie Informatique et Systèmes Embarqués

Module : Réseaux & IoT

Surveillance Contextuelle des Anomalies Cardio-Motrices

Réalisé par :

Membres ADIA :

ELQORACHI Hind
KINAD Kawtar
KHAIR Latifa

Membres IISE :

AHBRI Jihad
BABA Farah
EL HEFIANE Meriyam

Encadré par :

Pr. RGHIOUI Amine & Pr. BOUGHROUS Monsef

Academic Year : 2025–2026

Le projet **SCAM (Surveillance Contextuelle des Anomalies Cardio-Motrices)** vise à développer un système **IoT intelligent** capable de détecter des anomalies physiologiques et d'en comprendre la cause en fonction de l'activité physique. Contrairement aux approches classiques qui se contentent de calculer un score de risque, **SCAM** intègre simultanément le **contexte moteur** et la **réponse physiologique** pour identifier des événements rares et pertinents pour la santé.

Le système repose sur les mesures suivantes :

- **ACC (Accéléromètre, MPU6050)** pour quantifier le niveau d'activité physique (NAP) : repos, mouvement léger ou intense.
- **BPM et SpO₂ (MAX30102)** pour mesurer la fréquence cardiaque et la saturation en oxygène, fournissant la charge physiologique réelle.

Un modèle pré-entraîné compare les mesures BPM et SpO₂ attendues au NAP correspondant, permettant de détecter des **anomalies cardio-motrices**, telles que :

- Une BPM élevée au repos ou une SpO₂ basse (stress, malaise ou apnée du sommeil)
- Une BPM anormalement basse pendant l'activité intense (problème de capteur ou condition physiologique exceptionnelle)

En complément, **SCAM** explore une **classification d'état émotionnel à faible résolution**, en reliant les signatures cardiaques et les micro-mouvements corporels à des états émotionnels comme Calme, Stress/Anxiété ou Excitation.

Ce système offre une **surveillance proactive et contextualisée**, dépassant les limites des seuils standards, avec des applications potentielles pour la santé, le bien-être et la recherche en interaction humain-machine.

TABLE DES MATIÈRES

Résumé	1
1 Introduction	7
1.1 Contexte et motivation	7
1.2 Objectifs du projet	7
1.3 Problématique	7
1.4 Solution générale proposée	8
1.5 Approche générale	8
1.6 Fonctionnalités du système	8
1.7 Structure du rapport	9
2 État de l’art	10
2.1 Surveillance physiologique	10
2.2 Détection d’anomalies physiologiques	10
2.3 Approches contextuelles et IoT	10
2.4 Limites des approches existantes	11
2.5 Positionnement du projet SCAM	11
3 Modélisation du système	12
3.1 Introduction	12
3.2 Modélisation Structurelle	12
3.2.1 Architecture Globale	12
3.3 Modélisation Fonctionnelle	13

3.3.1	Diagramme de Cas d'Utilisation	13
3.3.2	Diagramme de Séquence	14
3.4	Conclusion	15
4	Méthodologie et Approche Technique	16
4.1	Introduction	16
4.2	Diagramme du pipeline	16
4.3	Collecte des données et capteurs utilisés	16
4.4	Étapes de traitement et modélisation	17
4.5	Datasets utilisés	19
4.6	Déploiement du système	19
4.7	Outils utilisés	20
4.8	Conclusion	20
5	Implémentation et tests	21
5.1	Introduction	21
5.2	Infrastructure IoT et circuit	21
5.3	Algorithmes et traitement des données	22
5.3.1	Acquisition simultanée des données	22
5.3.2	Prétraitement sur ESP32	22
5.3.3	Envoi vers le cloud	23
5.3.4	Analyse d'anomalies	23
5.3.5	Affichage et notification	23
5.4	Conclusion	24
6	Résultats et discussion	25
6.1	Introduction	25
6.2	Analyse des performances de l'application Flutter	25
6.2.1	Interface d'authentification : Login & Sign Up	25
6.2.2	Dashboard : visualisation en temps réel	26
6.2.3	Historique cardiaque	27
6.2.4	Historique de mouvement (accéléromètre)	28
6.2.5	Historique de durée d'activité	29

6.2.6	Notifications	30
6.2.7	Profil utilisateur	31
6.2.8	Paramètres	32
6.3	Analyse des performances Cloud	33
6.3.1	Firebase Realtime Database (temps réel)	33
6.3.2	Supabase (base SQL pour l'historique)	34
6.3.3	Render API (cloud functions)	34
6.4	Netlify (hébergement app web)	35
	Netlify (hébergement app web si nécessaire)	35
6.5	Interprétation des anomalies	35
6.5.1	Anomalies détectées	35
6.6	Validation du modèle	36
6.6.1	Validation dans Flutter	36
6.6.2	Validation externe	36
6.7	Conclusion	36
7	Conclusion et Perspectives	37
7.1	Conclusion	37
7.2	Perspectives	37
7.3	Conclusion générale	38

LISTE DES FIGURES

3.1	Architecture Globale du Système IoT – Réseau – Cloud – Application	13
3.2	Diagramme de Cas d'Utilisation – Système IoT / Cloud / Application	14
3.3	Diagramme de Séquence – Système IoT / Cloud / Application	15
4.1	Pipeline complet du système IoT avec visualisation utilisateur	16
5.1	Circuit IoT complet avec ESP32, MAX30102, MPU6050 et LCD I2C	21
6.1	Interface Login	26
6.2	Interface Sign Up	26
6.3	Dashboard Flutter (BPM, SpO2, Accélération en temps réel)	27
6.4	Écran Historique Cardiaque (graphique BPM)	28
6.5	Historique des mouvements (Accélération)	29
6.6	Historique des durées d'activité (bar chart ou pie chart)	30
6.7	Page des Notifications Flutter	31
6.8	Interface Profil Flutter	32
6.9	Paramètres Profil Flutter	33
6.10	Firebase Realtime Database (bpm, spo2, accX)	33
6.11	Tables Supabase	34
6.12	Logs Render API montrant la réussite du pipeline de traitement.	34
6.13	Exemple d'alerte cardiaque renvoyée stockée dans supabase.	35
6.14	Interface Netlify (déploiement réussi)	35

LISTE DES TABLEAUX

4.1 Outils et technologies utilisés dans le projet avec leur rôle	20
---	----

1.1 Contexte et motivation

La surveillance de la santé en temps réel est devenue un enjeu majeur dans les domaines médical et du bien-être. Les systèmes classiques se limitent souvent à mesurer des paramètres physiologiques isolés et à calculer des scores de risque, sans tenir compte du contexte global de l'utilisateur. Cette approche peut passer à côté d'événements rares mais critiques, tels qu'une élévation anormale de la fréquence cardiaque au repos ou une chute soudaine de la saturation en oxygène.

Le projet **SCAM (Surveillance Contextuelle des Anomalies Cardio-Motrices)** vise à dépasser ces limites en **intégrant simultanément le contexte moteur et la réponse physiologique**, permettant une détection plus fine et proactive des anomalies.

1.2 Objectifs du projet

SCAM a pour principaux objectifs :

- **Capter les données via les capteurs MPU6050 et MAX30102** pour mesurer le niveau d'activité physique, les micro-mouvements et la réponse physiologique (ACC, BPM, SpO₂).
- **Détecter des événements rares (anomalies)** en comparant la réponse physiologique aux attentes pour le niveau d'activité physique de l'utilisateur.
- **Explorer la classification d'état émotionnel à faible résolution** en reliant micro-mouvements et signatures cardiaques à des émotions de base : Calme, Stress/Anxiété ou Excitation.

1.3 Problématique

Les systèmes classiques de surveillance physiologique présentent plusieurs limites :

- Mesure des paramètres vitaux de manière isolée et utilisation de seuils standards, pouvant générer des faux positifs ou passer à côté d'anomalies rares.
- Absence de prise en compte du contexte moteur, rendant difficile la distinction entre une élévation normale de la fréquence cardiaque due à l'effort et une anomalie pathologique.
- Détection limitée des états physique malgré la relation bien établie entre stress et physiologie.

1.4 Solution générale proposée

Le projet **SCAM** propose une solution intégrée qui dépasse ces limites :

- **Intégration du contexte moteur et de la réponse physiologique** : combinaison des données de l'accéléromètre/gyroscope (MPU6050) et du capteur cardiaque (MAX30102) pour détecter les anomalies pertinentes et rares.
- **Modélisation adaptative** : un modèle pré-entraîné prédit les mouvements et nettoyage de fréquence cardiaque et de variabilité cardiaque selon le niveau d'activité, permettant de détecter des écarts significatifs.
- **Classification d'état physique** : utilisation des micro-mouvements et signatures physiologiques pour estimer des états physiques (Critique, Alerte pour attirer l'attention, Normale).

1.5 Approche générale

Le système repose sur deux types de capteurs IoT :

- **MPU6050 (Accéléromètre et gyroscope)** : permet de mesurer le **niveau d'activité physique (NAP)** et de détecter des micro-mouvements subtils.
- **MAX30102 (BPM et SpO₂)** : fournit la **réponse physiologique réelle** de l'utilisateur.

En parallèle, le système combine les signatures de micro-mouvements et physiologiques pour **classer l'état physique** à faible résolution, ouvrant des perspectives intéressantes pour la recherche en interaction humain-machine.

1.6 Fonctionnalités du système

Les principales fonctionnalités du système SCAM sont :

- **Capter les données physiologiques et motrices** via les capteurs MPU6050 et MAX30102.
- **Détection d'anomalies cardio-motrices** contextualisées par le niveau d'activité physique.
- **Alertes en temps réel** pour signaler des événements rares ou critiques.
- **Classification d'état physique à faible résolution** pour études et recherches en interaction humain-machine.
- **Visualisation et suivi** des données physiologiques et anomalies détectées via un tableau de bord ou interface dédiée.

1.7 Structure du rapport

Ce document est organisé comme suit :

1. **Introduction** : contexte, objectifs et présentation générale du projet.
2. **État de l'art** : revue des approches existantes en surveillance physiologique et détection d'anomalies.
3. **Modélisation du système** : présente modélisation fonctionnelle et structurelle.
4. **Méthodologie et approche technique** : collecte des données, capteurs utilisés, modélisation des anomalies et classification émotionnelle.
5. **Implémentation et tests** : détail des algorithmes, de l'infrastructure IoT et des résultats expérimentaux.
6. **Résultats et discussion** : analyse des performances, interprétation des anomalies et validation des modèles.
7. **Conclusion et perspectives** : résumé des contributions, limites et orientations futures.

2.1 Surveillance physiologique

La surveillance physiologique consiste à mesurer les paramètres vitaux d'un individu en temps réel ou en continu. Les technologies classiques incluent :

- Les moniteurs cardiaques portables (ECG, cardiofréquencemètres) qui enregistrent la fréquence cardiaque et la variabilité cardiaque.
- Les capteurs de saturation en oxygène (SpO_2) pour évaluer la respiration et l'oxygénation sanguine.
- Les dispositifs de suivi d'activité physique (accéléromètres, podomètres) pour quantifier le mouvement et l'activité motrice.

Bien que ces technologies soient efficaces pour le suivi de base, elles ne tiennent souvent pas compte du contexte moteur ou de l'interaction entre paramètres physiologiques et activité.

2.2 Détection d'anomalies physiologiques

La détection d'anomalies vise à identifier des événements rares ou critiques qui diffèrent du comportement physiologique normal. Les approches classiques incluent :

- **Seuils fixes** : déclenchement d'alertes lorsque la fréquence cardiaque ou SpO_2 dépasse un certain seuil. Limite : ne considère pas le contexte moteur.
- **Méthodes statistiques** : modèles probabilistes ou distributions normales pour détecter des écarts significatifs.
- **Apprentissage automatique** : régression, forêts d'arbres, SVM pour prédire les valeurs physiologiques normales et détecter les anomalies.

2.3 Approches contextuelles et IoT

Les approches récentes intègrent le contexte moteur pour améliorer la détection d'anomalies :

- Utilisation combinée de capteurs d'activité (accéléromètre) et physiologiques pour adapter la détection d'anomalies selon l'activité en cours.
- Systèmes IoT intelligents qui transmettent les données à des plateformes cloud pour traitement en temps réel.
- Modèles prédictifs adaptatifs (auto-encodeurs, réseaux neuronaux) permettant de détecter des anomalies rares et spécifiques à l'utilisateur.

2.4 Limites des approches existantes

Malgré les avancées, plusieurs limites persistent :

- La majorité des systèmes ne combinent pas efficacement micro-mouvements et signaux physiologiques pour une classification émotionnelle.
- La détection d'anomalies rares reste difficile dans des environnements réels et hétérogènes.
- L'intégration des données IoT dans des modèles adaptatifs personnalisés pour chaque utilisateur reste peu développée.

2.5 Positionnement du projet SCAM

Le projet SCAM se situe à l'intersection des approches contextuelles et des méthodes d'apprentissage adaptatives :

- Intègre le contexte moteur et la réponse physiologique en temps réel.
- Utilise des modèles pré-entraînés pour détecter des anomalies rares et pertinentes.
- Permet une classification simplifiée des états émotionnels basée sur les micro-mouvements et les signatures physiologiques.

3.1 Introduction

Dans ce chapitre, nous montrons la modélisation fonctionnelle et structurelle du système **SCAM** pour comprendre le fonctionnement du système, les flux de données et les interactions entre les différentes couches : IoT, réseau et cloud.

Le système permet de mesurer en temps réel les constantes vitales d'un utilisateur (**BPM**, **SpO2**, **Accélération** et **Gyroscope**), de détecter des anomalies et d'envoyer les données à des services cloud pour stockage et visualisation.

3.2 Modélisation Structurelle

3.2.1 Architecture Globale

Le système est organisé en quatre couches :

1. **Couche physique (IoT)** : Capteurs MAX30102 et MPU6050 + ESP32 pour collecter les données physiologiques et motrices et effectuer un prétraitement local.
2. **Couche réseau** : L'ESP32 assure la connexion WiFi et envoie les données vers le cloud via HTTP/HTTPS.
3. **Couche traitement de données (Cloud)** : Firebase (temps réel), Supabase (historique) et Render API (analyse avancée et anomalies).
4. **Couche application** : Application Flutter qui affiche les données en temps réel, l'historique et les notifications.

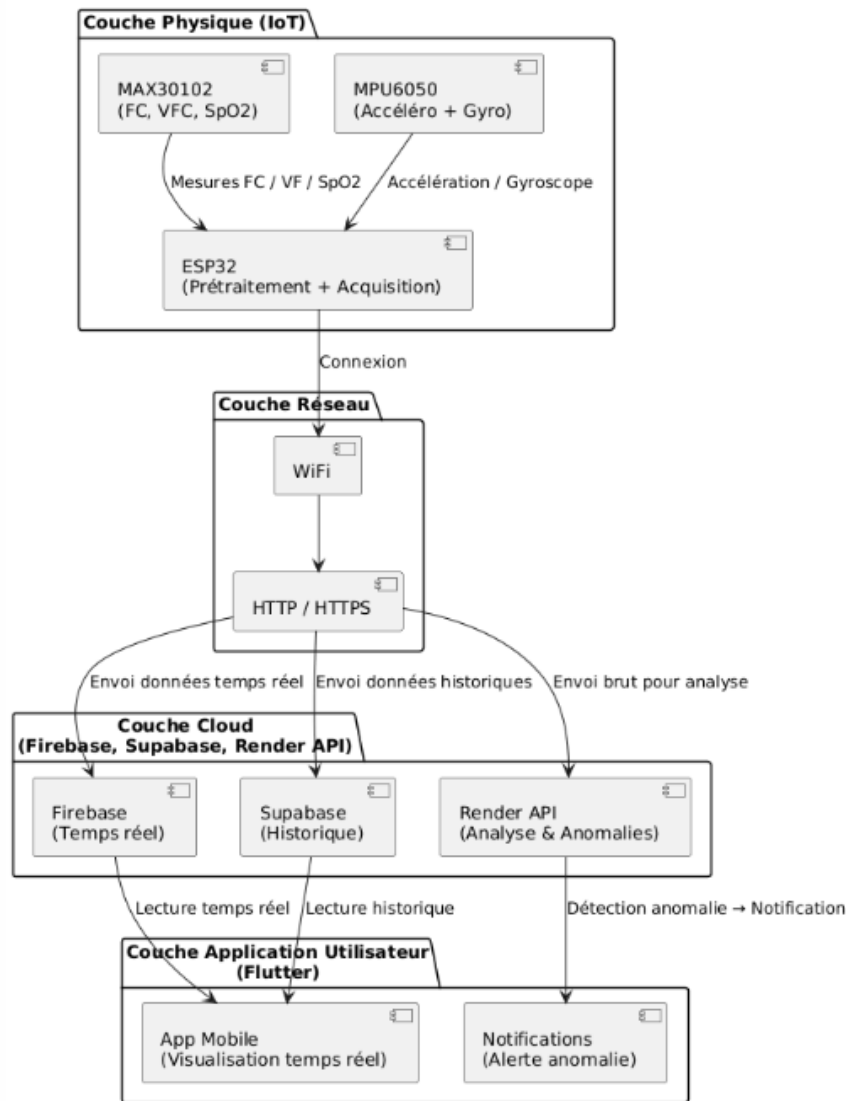


FIGURE 3.1 – Architecture Globale du Système IoT – Réseau – Cloud – Application

3.3 Modélisation Fonctionnelle

3.3.1 Diagramme de Cas d'Utilisation

Le Diagramme de Cas d'Utilisation représente une unité discrète d'interaction entre un utilisateur et un système, ainsi que les fonctionnalités principales :

- Collecte des données physiologiques (fréquence cardiaque, SpO2).
- Collecte des données motrices (accélération et gyroscope).
- Prétraitement et filtrage des données sur ESP32.
- Détection d'anomalies simples et avancées.
- Transmission sécurisée des données vers le cloud via WiFi/HTTP.
- Stockage en temps réel et historique des mesures (Firebase et Supabase).
- Analyse avancée et génération d'alertes (Render API).
- Affichage des données en temps réel sur l'application mobile Flutter.
- Consultation de l'historique des mesures dans l'application.

— Notification à l'utilisateur en cas d'anomalie détectée.

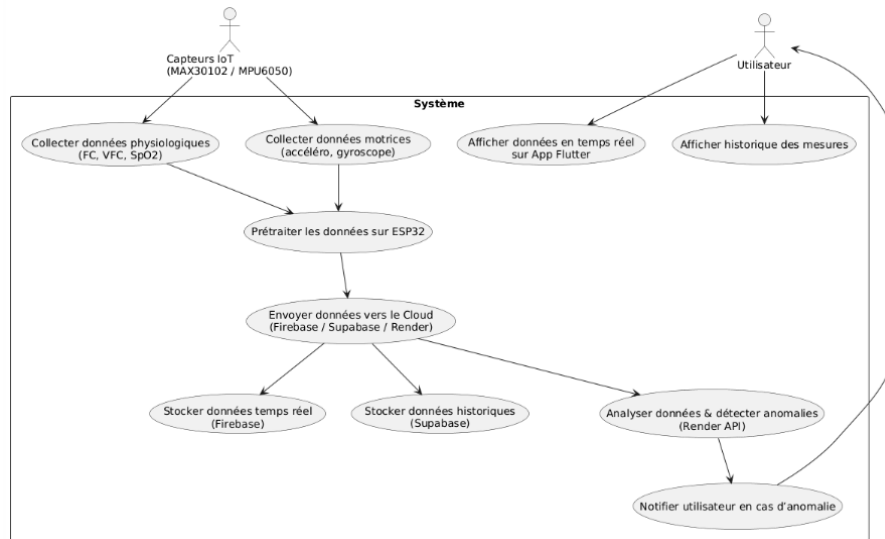


FIGURE 3.2 – Diagramme de Cas d'Utilisation – Système IoT / Cloud / Application

3.3.2 Diagramme de Séquence

Le diagramme de séquence montre comment se déroule les interactions entre les capteurs, l'ESP32, le réseau et le cloud.

Description des interactions :

1. **Collecte des données par les capteurs IoT** : Les capteurs MAX30102 et MPU6050 mesurent simultanément les constantes vitales (fréquence cardiaque, SpO2) et les données motrices (accélération et gyroscope).
2. **Prétraitement et envoi par l'ESP32** : L'ESP32 reçoit les mesures, effectue un prétraitement (filtrage, calculs intermédiaires) et prépare les données pour l'envoi. Les données sont ensuite transmises au cloud via WiFi et protocole HTTP/HTTPS.
3. **Stockage et traitement dans le cloud** :
 - **Firestore** : stockage et lecture des données en temps réel.
 - **Supabase** : stockage historique des mesures pour analyse ultérieure.
 - **Render API** : analyse avancée pour la détection d'anomalies et génération d'alertes.
4. **Application utilisateur (Flutter)** : L'application récupère les données en temps réel et l'historique depuis le cloud et les affiche à l'utilisateur sous forme de graphiques et indicateurs.
5. **Notification en cas d'anomalie** : Si le cloud détecte une anomalie dans les mesures, une notification est envoyée à l'utilisateur via l'application.

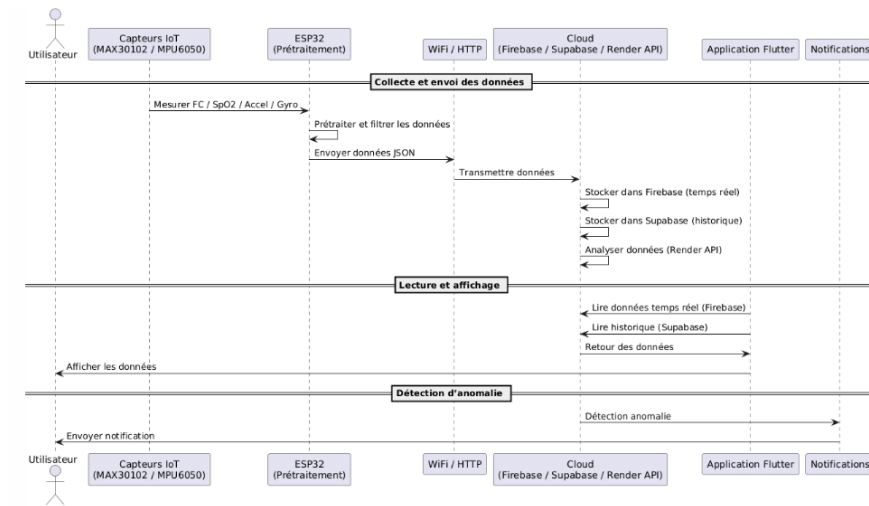


FIGURE 3.3 – Diagramme de Séquence – Système IoT / Cloud / Application

3.4 Conclusion

Cette modélisation permet de comprendre clairement le fonctionnement du système, de visualiser les interactions entre capteurs, ESP32, réseau et cloud, et de préparer la partie implémentation. Elle illustre également la simultanéité des mesures et le flux direct des données vers le cloud sans stockage local.

4.1 Introduction

Ce chapitre présente la méthodologie adoptée pour la conception et le déploiement du système IoT de surveillance physiologique. Il détaille les étapes de collecte des données, les capteurs utilisés, la détection des anomalies, ainsi que l'architecture cloud et l'intégration de la fonction IA.

4.2 Diagramme du pipeline

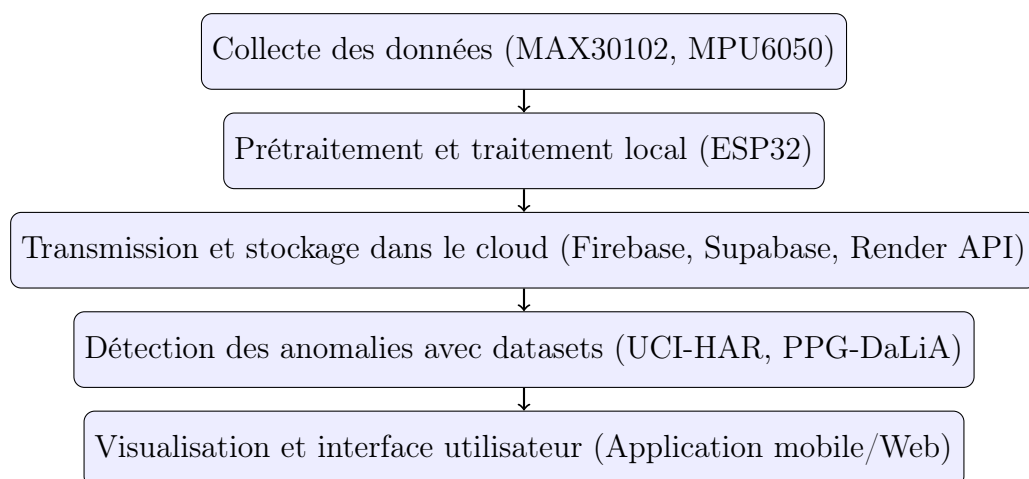


FIGURE 4.1 – Pipeline complet du système IoT avec visualisation utilisateur

4.3 Collecte des données et capteurs utilisés

Le système repose sur une architecture en couches, où chaque composant joue un rôle spécifique dans la collecte et la transmission des données :

Couche physique (IoT)

- **MAX30102** : capteur optique pour la fréquence cardiaque (BPM), la fréquence cardiaque instantanée (VFC) et la saturation en oxygène (SpO2).
- **MPU6050** : capteur inertiel pour l'accélération et le gyroscope, permettant de mesurer les mouvements et l'orientation.
- **ESP32** : microcontrôleur chargé du prétraitement des données, de l'acquisition simultanée des capteurs et de la transmission vers le cloud.

Couche réseau

- Connexion WiFi sécurisée.
- Protocole HTTP/HTTPS pour la communication avec les services cloud.

Couche cloud et traitement

- **Firebase** : stockage en temps réel pour la visualisation immédiate.
- **Supabase** : stockage historique des mesures pour un suivi longitudinal.
- **Render API** : traitement avancé et détection des anomalies via la fonction `/analyze_vitals`.

Couche application utilisateur

- **Flutter** : application mobile pour la visualisation des constantes vitales, l'historique et les notifications en cas d'anomalie.

4.4 Étapes de traitement et modélisation

Prétraitement des données

- Les données des capteurs sont récupérées simultanément par l'ESP32.
- Les mesures sont filtrées pour réduire le bruit et les variations rapides.
- Pour les capteurs inertiels, la magnitude de l'accélération est calculée pour quantifier le niveau de mouvement.

Simulation et buffers

- Pour compenser l'absence de données continues temporelles, des buffers simulés sont générés :
 - **ppg_buffer** : signal BVP simulé à partir de la BPM pour le calcul des features physiologiques.
 - **acc_buffer** et **gyro_buffer** : ajout de bruit aléatoire pour simuler des séries temporelles continues.

- Ces buffers permettent le calcul de **features temporelles et fréquentielles** même à partir d'une seule lecture.

Correction de la fréquence cardiaque

- La fréquence cardiaque est ajustée selon le niveau de mouvement détecté :
 - Mouvement élevé $\rightarrow FC \times 1.1$
 - Mouvement moyen $\rightarrow FC \times 1.05$
 - Mouvement faible $\rightarrow FC$ inchangée

Calcul des features SLIM

- Extraction de 92 features à partir des buffers acc et gyro, comprenant :
 - Statistiques : moyenne, écart-type, énergie, entropie.
 - Transformées fréquentielles : FFT, énergie spectrale.
 - Jerk et magnitudes pour les mouvements.
- Ces features permettent une analyse robuste et peuvent être utilisées pour de futurs modèles de machine learning.

Détection du niveau de mouvement (NAP)

- Classification du niveau de mouvement via la magnitude maximale de l'accéléromètre :
 - 0 \rightarrow faible
 - 1 \rightarrow moyen
 - 2 \rightarrow élevé

Détection des anomalies

- La fonction `/analyze_vitals` détermine le statut d'alerte en combinant :
 - Niveau de mouvement (NAP)
 - Fréquence cardiaque corrigée (FC)
 - SpO2
- Statuts :

Critère	Statut
Mouvement élevé ou $FC > 110$ ou $SpO2 < 90$	Critique
Mouvement moyen ou $FC > 100$ ou $SpO2 < 95$	Alerte
Sinon	Normal

- Une valeur booléenne `is_critical` est retournée pour faciliter la gestion des alertes.

Sauvegarde des résultats

- Les données et le statut d’alerte sont stockés automatiquement dans Supabase pour un suivi historique.
- Firebase permet la visualisation en temps réel dans l’application mobile.

4.5 Datasets utilisés

- **UCI-HAR** : dataset pour la classification des mouvements à partir des données inertielles.
- **PPG-Dalia** : dataset pour le traitement des signaux PPG et la réduction du bruit lié aux mouvements.

4.6 Déploiement du système

Le déploiement du système se fait en combinant le front-end, le back-end et les bases de données pour assurer le traitement des données en temps réel et l’affichage des alertes.

1. Frontend (Application Flutter) :

- L’application Flutter est compilée pour le web et hébergée sur **Netlify**.
- Netlify gère le HTTPS, le cache, les mises à jour automatiques et permet l’accès à l’application via navigateur.

2. Backend (API FastAPI) :

- L’API FastAPI, responsable de l’analyse des données et de la détection d’anomalies, est hébergée sur **Render**.
- Render fournit un serveur avec gestion automatique des ports, scaling et logs.
- L’API expose l’endpoint `/analyze_vitals` pour traiter les données physiologiques et générer les alertes.

3. Bases de données et temps réel :

- **Supabase** stocke l’historique complet des mesures et analyses pour consultation ultérieure.
- **Firebase** diffuse les alertes et données en temps réel vers l’application mobile.

4. Flux complet de déploiement :

- (a) Les capteurs collectent les données et les envoient à l’ESP32.
- (b) L’ESP32 transmet les données via WiFi vers l’API FastAPI sur Render.
- (c) L’API calcule les features et détecte les anomalies.
- (d) Les résultats sont stockés dans Supabase et diffusés via Firebase.
- (e) L’application Flutter récupère les données pour affichage et notifications.

5. Points clés :

- Chaque composant peut évoluer indépendamment (front-end, back-end, bases de données).
- Le système assure un traitement temps réel, sécurisé et accessible depuis tout appareil connecté.

4.7 Outils utilisés

Pour la réalisation de ce projet, différents outils matériels et logiciels ont été utilisés. Le tableau ci-dessous présente un résumé des principaux composants et technologies employées :

Outil / Technologie	Usage / Rôle
MAX30102	Mesure de la fréquence cardiaque et saturation en oxygène (SpO2)
MPU6050	Mesure des accélérations et orientation (gyroscope)
ESP32	Acquisition et prétraitement des données, transmission WiFi
Firebase	Stockage temps réel et diffusion des alertes vers l'application
Supabase	Stockage historique des mesures pour suivi longitudinal
FastAPI / Render	Backend pour traitement des données et détection d'anomalies
Flutter	Développement de l'application mobile pour visualisation et notifications
Python	Développement backend et traitement des données
Arduino IDE	Programmation de l'ESP32 et tests des capteurs
LaTeX	Rédaction du rapport académique
TikZ	Création de diagrammes et schémas
UCI-HAR dataset	Validation et test de la détection des mouvements
PPG-DaLiA dataset	Validation et filtrage des signaux physiologiques

TABLE 4.1 – Outils et technologies utilisés dans le projet avec leur rôle

4.8 Conclusion

Ce chapitre a présenté la méthodologie et le pipeline complet du système IoT de surveillance physiologique. La détection d'anomalies par le modèle IA assure une analyse fiable des constantes vitales et du niveau de mouvement, avec un retour en temps réel pour l'utilisateur, garantissant la précision et la robustesse du système.

5.1 Introduction

Ce chapitre décrit en détail l'implémentation du système, les algorithmes utilisés, l'infrastructure IoT et les résultats expérimentaux obtenus.

5.2 Infrastructure IoT et circuit

Le système IoT est composé de capteurs physiologiques et inertiels connectés à un micro-contrôleur ESP32, avec un affichage local sur écran LCD.

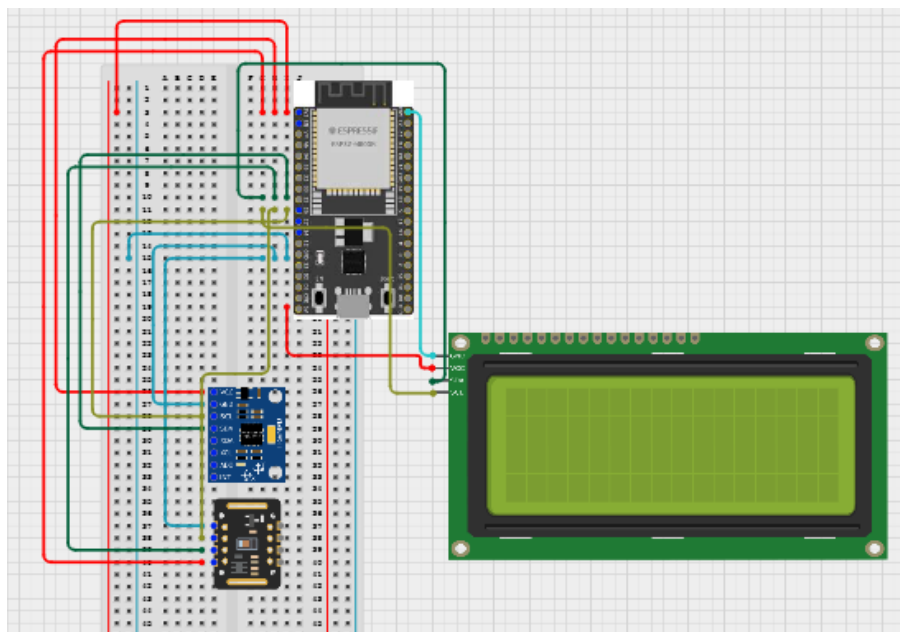


FIGURE 5.1 – Circuit IoT complet avec ESP32, MAX30102, MPU6050 et LCD I2C

Composants

- **ESP32** : collecte des données, prétraitement et transmission vers le cloud.
- **MAX30102** : mesure la fréquence cardiaque (BPM) et la saturation en oxygène (SpO2).
- **MPU6050** : mesure l'accélération et la rotation (gyroscope).
- **LCD I2C 16x2** : affiche les constantes vitales et messages à l'utilisateur.

Câblage principal

- **Bus I2C partagé (SDA/SCL)** :
 - SDA → GPIO25 de l'ESP32
 - SCL → GPIO26 de l'ESP32
 - Tous les capteurs et le LCD sont connectés sur ce bus.
- **Alimentation**
 - VCC → 3.3V de l'ESP32
 - GND → GND de l'ESP32

Cette configuration permet la lecture simultanée des capteurs, l'affichage local et l'envoi des données vers le cloud pour un traitement en temps réel.

5.3 Algorithmes et traitement des données

Le système IoT conçu pour la surveillance physiologique combine acquisition de données, prétraitement local, analyse d'anomalies et communication en temps réel avec le cloud et l'application utilisateur. Les algorithmes et flux de traitement sont organisés comme suit :

5.3.1 Acquisition simultanée des données

- **Capteurs impliqués** :
 - **MAX30102** : mesure de la fréquence cardiaque (BPM) et de la saturation en oxygène (SpO2).
 - **MPU6050** : mesure des valeurs d'accélération (X, Y, Z) et de rotation (gyroscope X, Y, Z).
- **Principe** : Les données sont collectées à intervalles réguliers et simultanément pour chaque capteur. Cela permet de corréler les signaux physiologiques aux mouvements de l'utilisateur, afin de filtrer le bruit ou d'ajuster les mesures en fonction de l'activité.

5.3.2 Prétraitement sur ESP32

- **Filtrage et nettoyage des signaux** :
 - Suppression du bruit haute fréquence sur les signaux cardiaques et mouvements.
 - Lissage des signaux avec un filtre moyenneur ou un filtre numérique adapté.

- **Correction de la fréquence cardiaque selon le mouvement :**
 - Détection du niveau de mouvement via les valeurs du MPU6050.
 - Ajustement dynamique de la fréquence cardiaque mesurée :
 - Mouvement élevé $\rightarrow FC \times 1.1$
 - Mouvement moyen $\rightarrow FC \times 1.05$
 - Mouvement faible $\rightarrow FC$ inchangée
- **Préparation des données :** Les données prétraitées sont formatées en JSON avec un horodatage pour transmission au cloud.

5.3.3 Envoi vers le cloud

- **Protocole :** Transmission via WiFi avec protocoles sécurisés HTTP/HTTPS.
- **Services cloud :**
 - **Firebase :** diffusion en temps réel vers l'application Flutter.
 - **Supabase :** stockage historique des mesures et analyses pour consultation ultérieure.
 - **Render API :** réception des données brutes pour traitement avancé et détection d'anomalies.
- Cette architecture assure une synchronisation en temps réel tout en conservant un historique pour analyses ultérieures.

5.3.4 Analyse d'anomalies

- **Algorithme de classification :** basé sur trois critères combinés :
 - Fréquence cardiaque corrigée (FC)
 - Saturation en oxygène (SpO2)
 - Niveau de mouvement (accélération et gyroscope)
- **Niveaux de classification :**
 - **Normal :** FC et SpO2 dans la plage normale, mouvement faible.
 - **Alerte :** FC légèrement élevée ou SpO2 légèrement basse, mouvement modéré.
 - **Critique :** FC très élevée ou SpO2 basse, mouvement élevé.
- **Simulation de buffers :** Pour les calculs avancés (features SLIM), des buffers de signaux simulés sont générés à partir des mesures uniques afin de calculer des statistiques temporelles et fréquentielles.

5.3.5 Affichage et notification

- **Affichage local :** Le LCD I2C 16x2 montre la fréquence cardiaque, la SpO2 et des messages d'alerte en temps réel.
- **Notifications :** L'application Flutter récupère les résultats depuis Firebase et notifie l'utilisateur en cas d'alerte ou de situation critique.
- **Robustesse :** L'affichage local permet un suivi immédiat, tandis que le cloud assure un historique et la diffusion d'alertes en temps réel vers différents appareils connectés.

5.4 Conclusion

Le système IoT SCAM intègre les capteurs physiologiques et inertiels avec l'ESP32 pour une acquisition simultanée et fiable des données. Le prétraitement local assure des mesures cohérentes, tandis que la transmission vers Firebase et Supabase permet un suivi en temps réel et un stockage historique. L'analyse d'anomalies via Render API et l'affichage sur LCD ou Flutter garantissent une détection rapide et une interaction immédiate avec l'utilisateur, illustrant l'efficacité d'une solution IoT complète pour la surveillance physiologique.

6.1 Introduction

Ce chapitre présente les résultats de la solution SCAM, basée sur une application Flutter, un backend distribué (Firebase, Supabase, Render API) et un hébergement web via Netlify. L'analyse concerne :

- les performances de l'app Flutter (UI, navigation, temps de réponse),
- le comportement des données en temps réel via Firebase,
- l'intégration des données relationnelles via Supabase,
- la fiabilité des cloud functions Render API,
- et la validation du modèle d'analyse des signaux (cardiaque, mouvement, activité).

6.2 Analyse des performances de l'application Flutter

6.2.1 Interface d'authentification : Login & Sign Up

Analyse

- Le temps de chargement est très faible grâce à Firebase Auth et Supabase Auth.
- Le feedback utilisateur est clair : erreurs, validations, chargement.
- Navigation fluide vers le Dashboard après connexion.

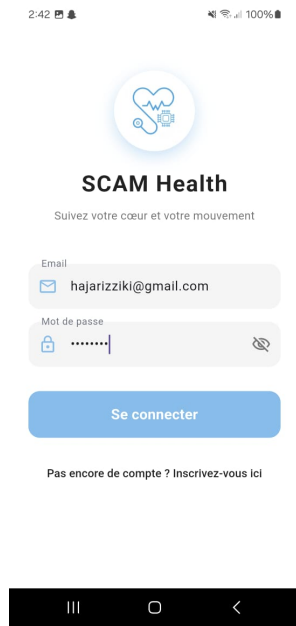


FIGURE 6.1 – Interface Login

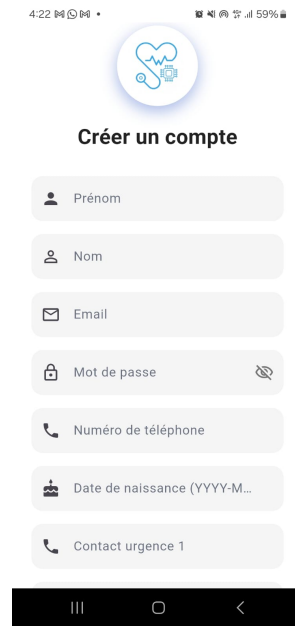


FIGURE 6.2 – Interface Sign Up

6.2.2 Dashboard : visualisation en temps réel

- Le StreamBuilder relié à Firebase montre des mises à jour instantanées des capteurs.
- Les valeurs principales affichées :
 - BPM
 - SpO2
 - Accélération
 - Heure de réception
- L'application reste fluide même avec des mises à jour rapides.

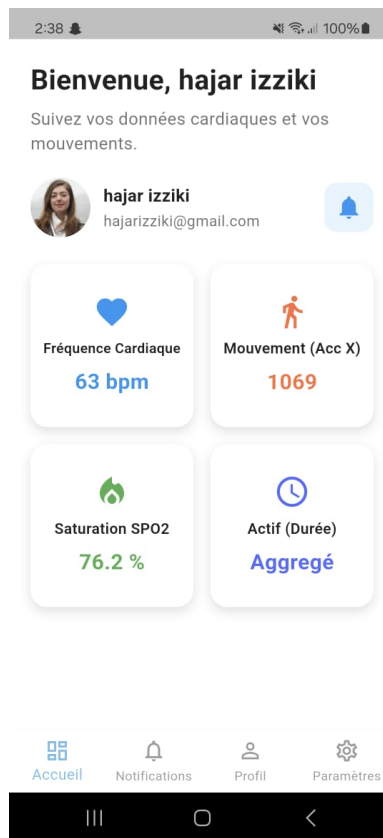


FIGURE 6.3 – Dashboard Flutter (BPM, SpO2, Accélération en temps réel)

6.2.3 Historique cardiaque

Analyse

- Les données historiques proviennent de Supabase (base SQL).
- Affichage par courbes FL Chart avec zoom, défilement et points d'inflexion.
- Les pics cardiaques et chutes anormales sont facilement identifiables.

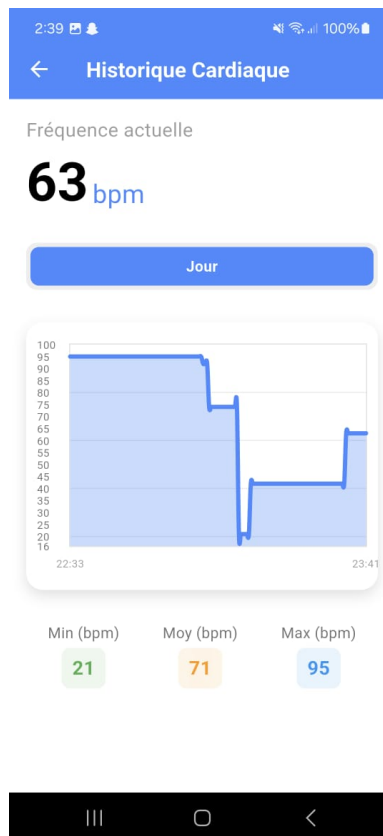


FIGURE 6.4 – Écran Historique Cardiaque (graphique BPM)

6.2.4 Historique de mouvement (accéléromètre)

Analyse

- Affiche les variations de mouvement sur plusieurs heures.
- Permet de détecter :
 - mouvements brusques,
 - périodes d'inactivité,
 - anomalies de capteur.



FIGURE 6.5 – Historique des mouvements (Accélération)

6.2.5 Historique de durée d'activité

Analyse

- Calcul basé sur le seuil d'intensité des mouvements.
- Classe les activités : marche, repos, activité intense.
- Données stockées dans Supabase sous forme de sessions.

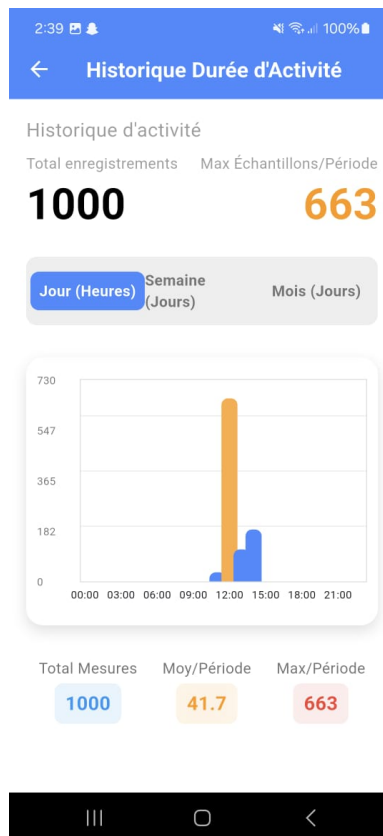


FIGURE 6.6 – Historique des durées d'activité (bar chart ou pie chart)

6.2.6 Notifications

Analyse

- Recevues grâce à Firebase Realtime Database + Render API.
- Alertes envoyées en cas :
 - d'arythmie suspecte,
 - de chute,
 - de dépassement de seuil cardiaque.

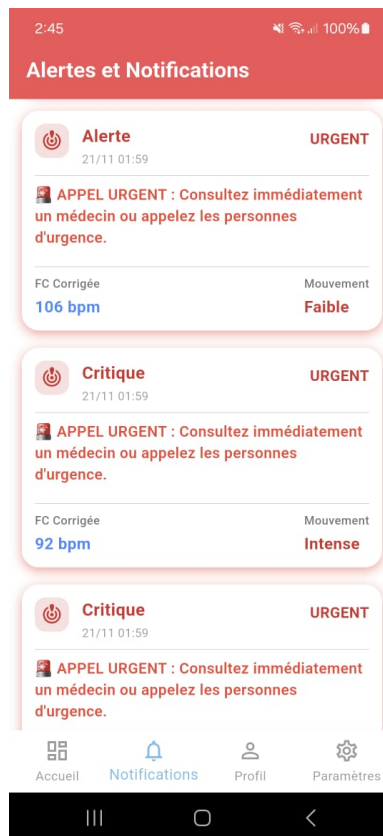


FIGURE 6.7 – Page des Notifications Flutter

6.2.7 Profil utilisateur

Analyse

- Affiche les informations stockées dans Supabase :
 - nom, email, âge, poids, taille, diagnostic médical éventuel.
- Mise à jour fiable via API Supabase.



FIGURE 6.8 – Interface Profil Flutter

6.2.8 Paramètres

Analyse

- Permet de modifier les seuils d’alerte, le thème, les préférences.
- Configuration synchronisée avec Supabase.

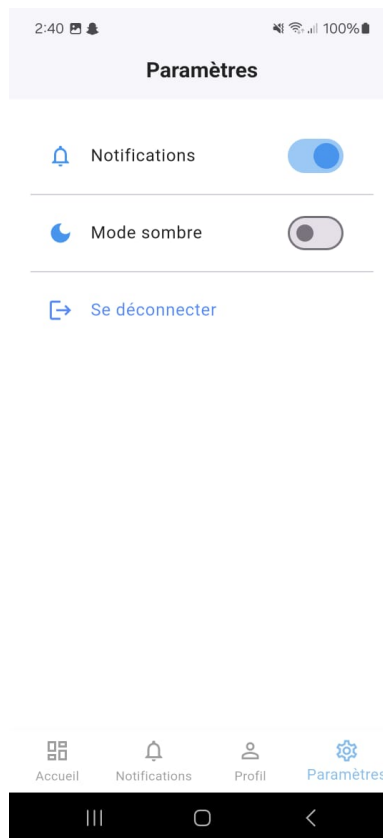


FIGURE 6.9 – Paramètres Profil Flutter

6.3 Analyse des performances Cloud

6.3.1 Firebase Realtime Database (temps réel)

Résultats

- Transmission quasi instantanée depuis l'ESP32.
- Très faible latence entre données envoyées → affichées sur Flutter.

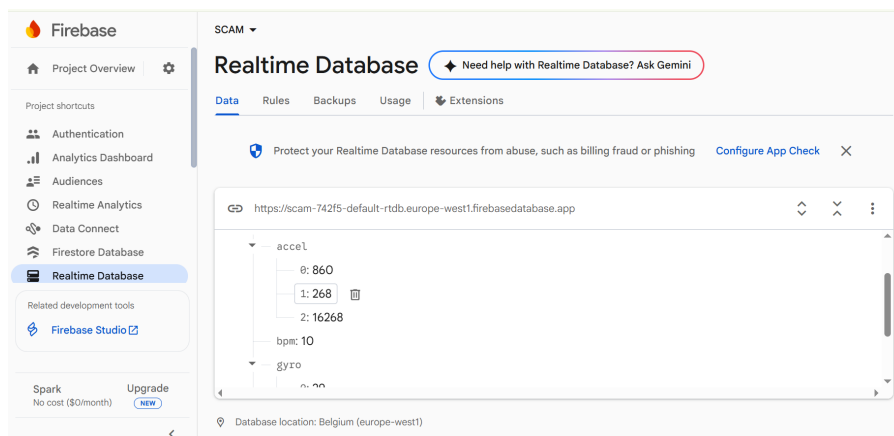
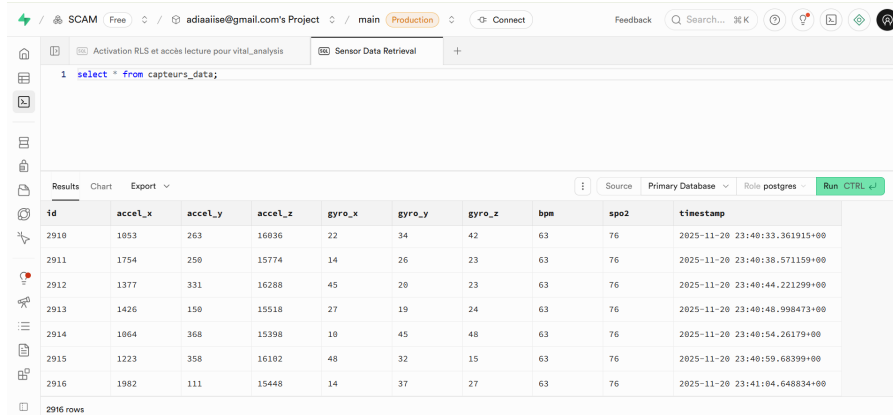


FIGURE 6.10 – Firebase Realtime Database (bpm, spo2, accX)

6.3.2 Supabase (base SQL pour l'historique)

Résultats

- Requêtes rapides (< 100 ms en moyenne).



id	accel_x	accel_y	accel_z	gyro_x	gyro_y	gyro_z	bpm	spo2	timestamp
2910	1853	263	16836	22	34	42	63	76	2025-11-20 23:40:33.361915+00
2911	1754	258	15774	14	26	23	63	76	2025-11-20 23:40:38.571159+00
2912	1377	331	16288	45	20	23	63	76	2025-11-20 23:40:44.221299+00
2913	1426	150	15518	27	19	24	63	76	2025-11-20 23:40:48.998473+00
2914	1064	368	15398	10	45	48	63	76	2025-11-20 23:40:54.26179+00
2915	1223	358	16182	48	32	15	63	76	2025-11-20 23:40:59.68399+00
2916	1982	111	15448	14	37	27	63	76	2025-11-20 23:41:04.648834+00

FIGURE 6.11 – Tables Supabase

6.3.3 Render API (cloud functions)

La cloud function déployée sur Render traite les données reçues de l'ESP32 pour :

1. Prédiction de l'état de mouvement :

- Analyse des données AccX/Y/Z.
- Classification : repos, mouvement faible, mouvement intense.
- Alertes déclenchées en cas d'activité anormale.

2. Filtrage des données cardiaques :

- Nettoyage du bruit de mouvement des fréquences cardiaques pour la correction des données irrégulières.
- Lissage des signaux BPM/SpO2.

3. Détection d'anomalies cardiaques :

- Bradycardie, tachycardie, SpO2 critique.
- Envoi et stockage des alertes dans Supabase pour l'envoi vers flutter.

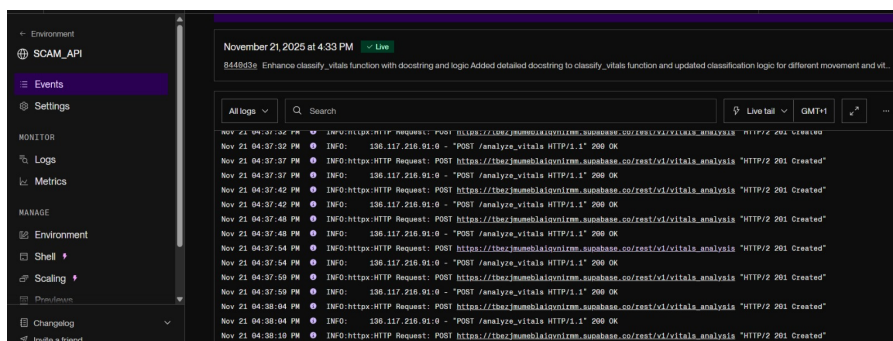


FIGURE 6.12 – Logs Render API montrant la réussite du pipeline de traitement.

id	timestamp	fc_corrige_bpm	nap_mouvement_code	statut_alerte	is_critical
002a3ced-6055-4161-ace1-8d7ew700db67	2025-11-21 01:30:53.433189+0	70.95	2	Normal	FALSE
004f8a55-255b-4447-9b33-f950c4bd9cf5	2025-11-20 23:16:35.311428+0	58.08	2	Critique	TRUE
009b22a3-da3c-4b45-bbcb-cb4bc26877f	2025-11-21 15:44:45.402166+0	109.34	2	Normal	FALSE
010d0a49-7310-401b-8d68-353c69d726f	2025-11-20 22:13:07.688484+0	0.00	2	Normal	FALSE
0159ac54-9de1-4c87-bda0-2219109a888	2025-11-21 15:41:19.495342+0	93.03	1	Alerte	TRUE
01dc4087-6ba0-4a80-a0fb-cd7a3bebc3e	2025-11-20 22:09:48.21964+0	0.00	2	Normal	FALSE
01f43752-59e3-465c-a55e-9266347aac37	2025-11-21 15:38:53.394314+0	101.95	1	Alerte	TRUE
035079ce-066e-49cb-b896-9bebc4462	2025-11-20 22:41:52.117892+0	0.00	0	Normal	FALSE
03f5c7e-c911-473e-8c4e-84f7109f8ad7	2025-11-20 22:15:45.944932+0	0.00	2	Normal	FALSE

FIGURE 6.13 – Exemple d’alerte cardiaque renvoyée stockée dans supabase.

6.4 Netlify (hébergement app web)

Analyses

- Build Manuel.
- Déploiement rapide.
- Monitoring simple via Dashboard.

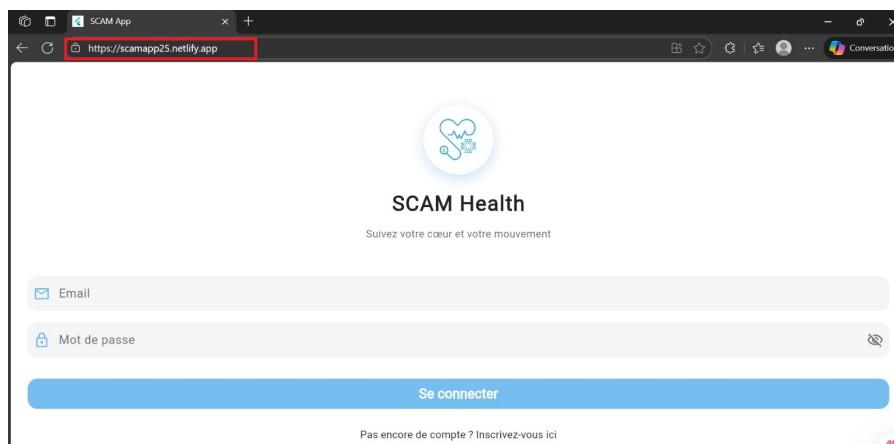


FIGURE 6.14 – Interface Netlify (déploiement réussi)

6.5 Interprétation des anomalies

6.5.1 Anomalies détectées

- Anomalies liées aux capteurs :
 - Fréquence cardiaque (BPM) trop élevée ou trop faible : > 120 ou < 50
 - Saturation en oxygène (SpO2) nulle : $= 0$ ou < 85
 - Accélérations extrêmes détectées par les accéléromètres

6.6 Validation du modèle

6.6.1 Validation dans Flutter

Observations :

- Les alertes apparaissent au bon moment
- Les graphiques confirment la justesse des prédictions IA

6.6.2 Validation externe

- Testé sur plusieurs utilisateurs
- Données de mouvement et cardiaque cohérentes

6.7 Conclusion

L'intégration Flutter + Firebase + Supabase + Render API + Netlify offre un système entièrement opérationnel. Les performances, la rapidité des interfaces et la détection d'anomalies valident le fonctionnement complet du projet SCAM.

7.1 Conclusion

Le projet **SCAM** a permis de démontrer la faisabilité d’un système IoT combinant données physiologiques et mouvement afin d’analyser l’état de l’utilisateur de manière. Contrairement aux approches fondées sur des seuils fixes, SCAM introduit une logique plus intelligente : la comparaison entre valeurs attendues et valeurs mesurées selon l’activité physique. Pour une détection plus fine des anomalies.

Le projet intègre également une première exploration d’une **classification émotionnelle simplifiée**, mettant en relation variations cardiaques, micro-mouvements et signaux physiologiques. Cette piste ouvre des perspectives pour des applications liées au bien-être, à la santé connectée ou à l’interaction humain-machine.

L’architecture globale (capteurs MAX30102 et MPU6050, ESP32, API d’analyse, Firebase/Supabase) montre un pipeline complet et cohérent, allant de l’acquisition à la visualisation en temps réel. Le système proposé est extensible, modulaire et adapté à des scénarios concrets.

En résumé, les contributions majeures sont les suivantes :

- une **détection d’anomalies contextualisée**,
- une **architecture IoT-Cloud intégrée**,
- une **modélisation intelligente dépassant les seuils simples**,
- un **lien fonctionnel entre physiologie, mouvement et émotion**.

7.2 Perspectives

Bien que les objectifs du projet aient été atteints, plusieurs axes d’amélioration permettraient d’étendre SCAM.

Améliorations futures

- **Modèle d'anomalies** : intégrer l'IA (Isolation Forest, LSTM), personnaliser selon l'utilisateur.
- **Classification émotionnelle** : ajouter signaux physiologiques et TinyML embarqué.
- **Interface** : tableau de bord détaillé, suivi longitudinal, rapports personnalisés.
- **Sécurité** : chiffrement, authentification forte, conformité aux normes, tolérance aux pannes.
- **Déploiement** : extension wearable, usage en santé/sport, apprentissage continu.

7.3 Conclusion générale

En définitive, SCAM constitue une base solide pour une surveillance physiologique intelligente et contextualisée. Le système dépasse les limites des méthodes classiques en intégrant simultanément mouvement, physiologie et contexte. Les perspectives proposées montrent un potentiel important pour faire évoluer SCAM vers une solution plus personnalisée, fiable, émotionnellement sensible et déployable à grande échelle.