



Référence : SPEC_PFE_VI

Version : 1.1

Révision : 4

Equipe : Sami KOUATLI
Achraf KORICHE
Axel ROLLO
Corentin LE DIZES

Etat du document : Fermé

Spécifications

Ville intelligente – Station qualité de l'air

Notes : Le présent document est un document à but pédagogique. Il a été réalisé sous le tutorat de Mr. PERDRIAU et Mr. RAMDANI, en collaboration avec des étudiants (Sami Kouatli, Axel Rollo, Corentin LE DIZES) de l'option SE et d'un étudiant (Achraf KORICHE) de l'option OC du Groupe ESEO.

Liste des évolutions et validations du document.

Date	Action	Auteur	Version	Révision
6 Octobre 2017	Premières ébauche des différentes parties	Axel ROLLO et Sami KOUATLI	1.0	1
18 Octobre 2017	Finalisation de la spécification dans sa première version	Sami KOUATLI	1.0	3
20 Octobre 2017	Modification de la spécification en parallèle de la conception	Sami KOUATLI	1.1	4

Table des matières

1 INTRODUCTION	5
1.1 Objet	5
1.2 Portée	5
1.3 Définitions, acronymes et abréviations	6
1.4 Références	7
1.5 Vue d'ensemble	8
2 DESCRIPTION GÉNÉRALE	9
2.1 Caractéristiques des acteurs	10
2.1.1 Acteurs directs	10
2.1.2 Acteurs hors champs	10
2.2 Environnement	11
2.2.1 Architecture matérielle et logicielle	11
2.2.2 Interfaces du système	16
2.2.2.1 Les interfaces logiques	16
2.2.2.2 Les interfaces avec les acteurs	17
2.2.2.2.1 Entre l'utilisateur et le E_Smartphone	17
2.2.2.2.1.1 En provenance de l'utilisateur	17
2.2.2.2.1.2 En provenance du E_Smartphone	17
2.2.2.2.2 Entre le E_Smartphone et le ServerStorage	18
2.2.2.2.2.1 En provenance du E_Smartphone	18
2.2.2.2.2.2 En provenance de StorageServer	18
2.2.2.2.3 Entre la Station et le Broker	18
2.2.2.2.3.1 En provenance de la Station	18
2.2.2.2.3.2 En provenance du Broker	18
2.2.2.2.4 Entre le Broker et StorageServer	18
2.3 Fonctions principales développées	19
2.3.1 Rappel sur le cas d'usage	19
2.3.2 Rappel sur le cas d'utilisation	19
2.3.3 CU Informer périodiquement les utilisateurs afin d'anticiper les risques de réactions allergiques	21
2.3.3.1 Description graphique	21
2.3.3.2 Description textuelle	21
2.3.4 CU Rafraîchir les données	23
2.3.4.1 Description graphique	23
2.3.4.2 Description textuelle	23
2.4 Contraintes	24
2.4.1 Politique réglementaire	24
2.4.2 Contraintes matérielles	24
2.4.3 Contraintes environnementales	24

2.4.4	Exigences de fiabilité	24
2.4.5	Exigences de maintenabilité	24
2.4.6	Exigences de disponibilité	24
2.5	Hypothèses et dépendances	25
3	EXIGENCES SPÉCIFIQUES	26
3.1	Interface Homme - Machine	26
3.1.1	Généralités	26
3.1.2	Action des utilisateurs	27
3.1.2.1	Diagramme d'enchaînement des vues de Poseidon	27
3.1.3	Les écrans	27
3.1.3.1	Écran de chargemennt	28
3.1.3.2	Écran d'accueil	29
3.1.3.3	Ecran de visualisation des données	30
3.1.3.4	Pop Up erreur de connexion	31
3.2	Dictionnaire du domaine	32
4	VALIDATION DU DOCUMENT	36

Chapitre 1

INTRODUCTION

1.1 Objet

Ce dossier de spécification a pour objectif de définir les fonctionnalités et exigences attendues par Mr. Perdriau selon le cahier des charges. Ce document va faire part de ces derniers pour l'élaboration du développement logiciel et matériel du Projet « Ville intelligente – Qualité de l'air » soutenu par la PAVIC.

Ce document permettra à l'équipe de développer les différentes cartes et de développer l'application Android.

L'élaboration du document est constituée grâce aux différents entretiens effectués avec Mr. Ramdani et Mr. Perdriau. Pour approuver notre compréhension du déroulement du projet, des réunions hebdomadaires ont été mises en place avec toute l'équipe afin de relever les interrogations soulevées sur le déroulement du projet.

Ce dossier de spécification suit les recommandations de la norme IEEE 830-1998 [IEEE 830_1998]. Il utilise des schémas et illustrations respectant la norme UML en version 2.5 [UML_2.5_2011].

1.2 Portée

Ce document décrit les fonctionnalités et exigences du Système à l'Etude (SaE) constitué :

- les logiciels « Athena », « Apollon », « Hermes » et « Zeus » sont embarqués sur les quatre modules « Thermos », « Luminos », « Aeros », « Luminos » et permettant d'effectuer les différents relevés météorologiques.
- du logiciel « Hades », embarqué sur la centrale, fais l'acquisition de des données GPS et visuelles. Elle permet aussi de mettre en forme toutes les données des modules et de les transmettre sur le réseau Wi-Fi.
- du logiciel « Poseidon » s'exécutant sur un support Android (tablette ou smartphone tournant sur Android). Ce dernier a pour objectif dans un premier temps, d'informer l'utilisateur avec des données météorologiques.
- du module « Thermos », comprenant les capteurs de température, de pression et d'humidité.

- du module « Aeros », comprenant les capteurs mesurant les différents gaz et de particules dans l'air.
- du module « Précipitos », comprenant la girouette, l'anémomètre et le pluviomètre.
- du module « Luminos », comprenant les capteurs liés au soleil, tel que les capteurs UV et de luminosité.
- de la « Nucleus », comprenant le GPS, la caméra et le module Wi-Fi permet de gérer le SaE afin de délivrer les informations sur une base de données.

Les modules et la centrale doivent piloter en activant et désactivant les capteurs du SaE. Des protocoles de communication et des caractéristiques spécifiques sont définis afin de permettre aux différents modules de communiquer avec la centrale et des noeuds de communiquer avec les différents capteurs. Ces caractéristiques seront décrites plus loin dans ce document.

1.3 Définitions, acronymes et abréviations

Les abréviations utilisées dans le présent document sont répertoriées et expliquées dans le tableau présenté ci-dessous. Les termes utiles pour interpréter correctement ce dossier de spécifications sont définis dans le dictionnaire de domaine présent dans ce dossier dans la partie 3.3.

Acronymes, abréviations	Définitions
API (Application Programming Interface)	Interface de programmation fournissant une liste de méthode et classes à un programme.
AQ (Assurance Qualité)	Assurance que tous les documents suivent le PAQL.
DB	Base De Données (DataBase).
CU	Cas d'utilisation
Disponibilité	La disponibilité est l'aptitude d'un composant ou d'un système à être en état de marche à un instant donné.
Fiabilité	La fiabilité est l'aptitude d'un composant ou d'un système à fonctionner pendant un intervalle de temps.
GPIO (General Purpose Input/Output)	Désigne des ports d'entrée/sortie
IEEE (Institute of Electrical and Electronics Engineers)	Association professionnelle internationale définissant, entre autres, des normes dans le domaine informatique et électronique.
IHM (Interface Homme Machine)	Moyen permettant à l'administrateur et aux emprunteurs d'interagir avec AutoShelves.
Maintenabilité	La maintenabilité est l'aptitude d'un composant ou d'un système à être maintenu ou remis en état de fonctionnement.
MQTT (MQ Telemetry Transport)	Protocole de messagerie publish-subscribe basé sur le protocole TCP/IP.

N.A	Non applicable
OMG (Object Management Group)	Association professionnelle internationale définissant, entre autres, des normes dans le domaine informatique.
PAQL (Plan d'Assurance Qualité Logicielle)	Dossier permettant d'assurer une qualité au projet et de faciliter ce dernier.
PAVIC	Plateforme Aménagement Ville Intelligente et Connectée (Angers).
RF	Radio fréquence, désigne les ondes radio dont le spectre est situé entre 3 kHz to 300 GHz.
SaE (Système à l'Etude)	Il s'agit de l'ensemble des composants de la station et du broker.
Sûreté de Fonctionnement (SdF)	La sûreté de fonctionnement est l'aptitude d'une entité à satisfaire à une ou plusieurs fonctions requises dans des conditions données.
TV	Tests de Validation
UI (User interface)	Interface utilisateur
UML (Unified Modeling Language)	Notation graphique normalisée, définie par l'OMG et utilisée en génie logiciel.
UX (User Experience)	Expérience utilisateur
Wi-Fi (Wireless Fidelity)	Protocole de communication sans fil régis par les normes du groupe IEEE 802.11.

TABLE 1.1 – Tableau des acronymes et abréviations

1.4 Références

Voici un tableau récapitulatif des documents utilisés pour ce document ainsi que les liens permettant d'accéder aux fichiers.

Référence	Nom, auteur
[IEEE-830_1998]	IEEE, std 830-1998 « Recommended Practice for Software Requirements Specifications », http://standards.ieee.org/findstds/standard/830-1998.html , 1998.
[UML_2.4_2011]	OMG, « Unified Modeling Language », version 2.5, http://www.omg.org/spec/UML/ , 2015.
[CDC_PFE_VI]	« Cahier des charges » pour le projet de station de qualité de l'air, version 1.00, Référentiel Documentaire, 2017.

[PAQL_PFE_VI]	Plan d'assurance qualité logiciel pour le projet de station de qualité de l'air, Référentiel Documentaire, 2017.
[MIM_PFE_VI]	Manuel d'installation et de maintenance pour le projet de station de qualité de l'air, Référentiel Documentaire, 2017.
[SPEC_PFE_VI]	Dossier de specifications générales pour le projet de station de qualité de l'air, fait référence à ce dossier, Référentiel Documentaire, 2017.
[CONC_LOG_PFE_VI]	Dossier de conception logicielle pour le projet de station de qualité de l'air, Référentiel Documentaire, 2017.
[CONC_MAT_PFE_VI]	Dossier de conception et de réalisation matérielle pour le projet de station de qualité de l'air, Référentiel Documentaire, 2017.

TABLE 1.2 – Tableau des références

1.5 Vue d'ensemble

Ce document est structuré en 3 parties :

- La partie 1 présente les objectifs et la portée de ce document.
- La partie 2, intitulée « Description Générale », a pour objectif de présenter l'environnement et le contexte de la station de qualité de l'air, ainsi que ses fonctionnalités principales attendues.
- La partie 3 présente en détail les IHM attendues, les spécifications des différents logiciels, les fonctionnalités détaillées ainsi que le dictionnaire du domaine.

Chapitre 2

DESCRIPTION GÉNÉRALE

L'association PAVIC a pour objectif de mettre en place une plateforme pour faciliter les réalisations d'expérimentations, contribuant à améliorer la vie des habitants d'une ville. Cette association contribue globalement à la mise en place de nouveaux dispositifs afin de développer les problématiques de ville intelligente.

PAVIC est le « living lab » qui va permettre à des entreprises de pouvoir tester leurs produits sur le territoire angevin. L'intérêt est d'avoir un prototype déployé à l'échelle local, qui est suffisamment grand mais aussi suffisamment petit pour que les lieux de décision soient proches. Par la suite, le produit pourra être développé dans la plupart des villes de France. Dans ce cadre, un certain nombre d'initiatives sont engagées par les industriels et les académiques pour réaliser ces prototypes, dont l'ESEO avec notre sujet.

Les allergies à diverses espèces de pollens et/ou la sensibilité à la pollution de l'air constituent une problématique importante pour une proportion significative de la population. Dans ce cadre, la prédition de l'arrivée, au niveau de la ville voire du quartier, d'une vague de pollen ou de pollution permettrait aux personnes allergiques et/ou asthmatiques d'anticiper les mesures de protection nécessaires.

Les objectifs sont donc de palier cette problématique, en apportant une courbe de prédition intuitive accessible par l'utilisateur depuis l'application « Poseidon ». La réalisation du projet peut se décomposer en plusieurs parties :

- Une première étape permettant d'établir des mesures météorologiques grâce aux différents capteurs.
- Une seconde étape consacrée à la centrale, capable d'acquérir et transmettre ces mesures vers un serveur privé ou public.
- Une troisième étape permettant de recevoir et stocker les données dans une base de données.
- Une dernière étape permettant d'envoyer et de visualiser les données sur une application Android.

2.1 Caractéristiques des acteurs

2.1.1 Acteurs directs

- **Utilisateur** : la personne utilisant l'application dans le but de connaître le taux de pollen et la pollution ambiante.
- **Broker** : Serveur MQTT qui fait le lien entre la Nucleus et le StorageServer.
- **StorageServer** : le serveur s'occupant de récupérer les données du Broker MQTT et stocke les données dans sa base de données.

2.1.2 Acteurs hors champs

- **Réglementation** : Norme IEEE 802.11 : Le smartphone communiquera avec le Broker par le biais du Wi-Fi et devra donc respecter la norme 802.11 régissant les réseaux sans-fil LAN de type Wi-Fi.
- **Réglementation** : Norme NF EN 50566 : Les modules communiqueront avec le Nucleus par le biais d'onde RF et devra donc respecter la norme NF EN 50566 régissant les réseaux champs radiofréquence.

2.2 Environnement

2.2.1 Architecture matérielle et logicielle

Le diagramme de déploiement UML de la figure 2.1 représente l'architecture logicielle et matérielle du SaE. Les conventions graphiques utilisées sont explicitées en figure 2.2.

Ce diagramme de déploiement identifie les entités matérielles et/ou logicielles avec lesquelles le SaE doit interagir et permet ainsi de déterminer les principaux échanges qu'il entretient avec son environnement.

L'ensemble des diagrammes suivants sont à titre purement indicatif, l'équipe de développement se réserve le droit d'exercer des modifications. Les composants choisis pour la mise en place du projet sont déterminés dans le dossier de conception et de réalisation matérielle [CONC_MAT_PFE_VI] livré avec le projet.

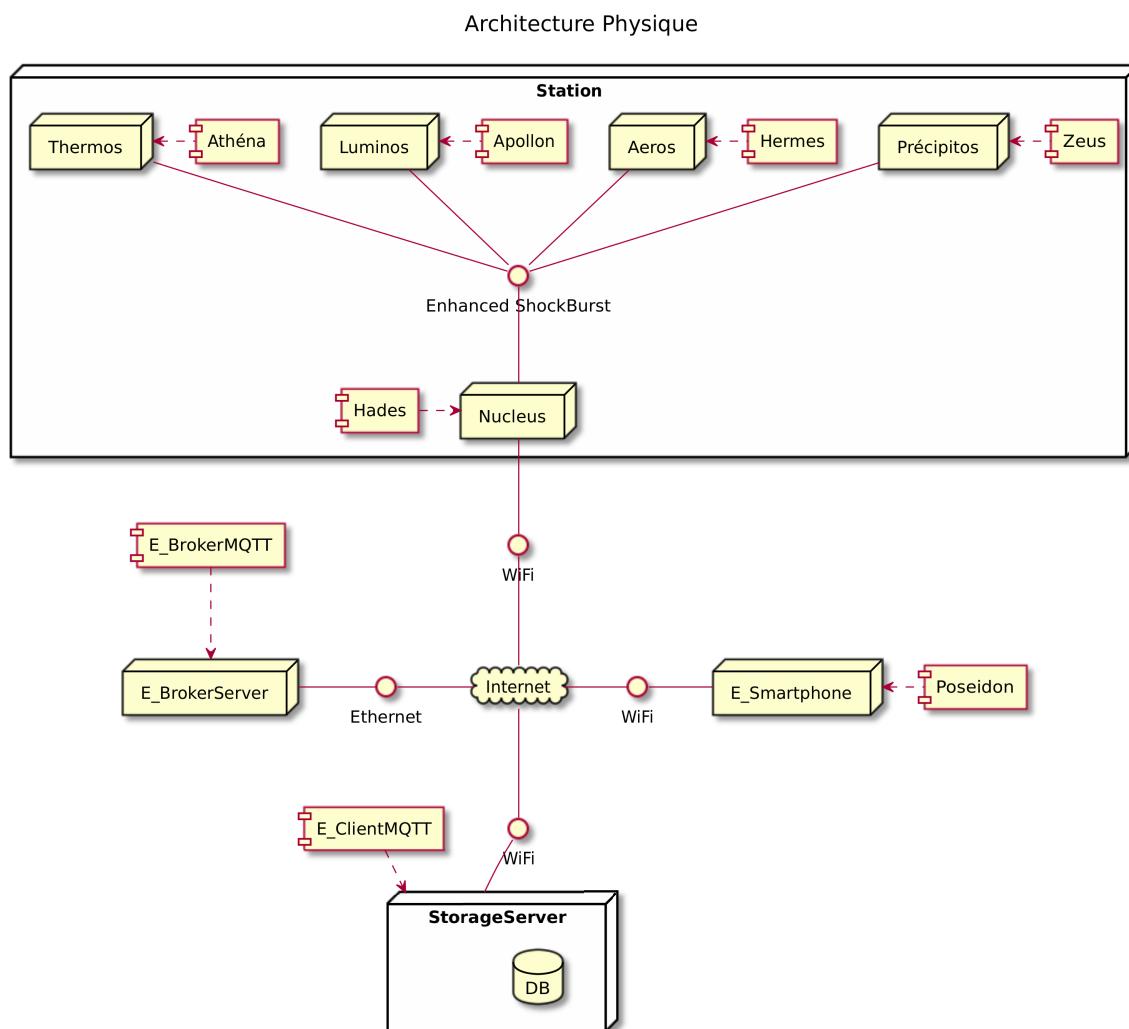


FIGURE 2.1 – Architecture matérielle et logicielle

Comme indiqué sur l'illustration ci-dessus, notre projet regroupe différentes entités. La station se compose de 4 noeuds de capteurs (Thermos, Luminos, Aeros et Precipitos) ainsi que d'une centrale Nucleus. Il interagit aussi avec une entité externe au système, le E_Smartphone de l'utilisateur. Cette entité est préfixé par les lettres «E_» (E pour Externe).

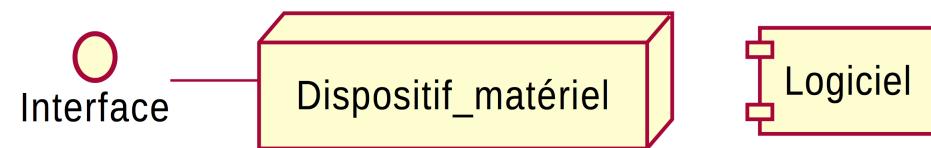


FIGURE 2.2 – Légende du diagramme de déploiement UML

Les capteurs des noeuds de capteurs récupèrent des informations sur l'environnement puis les envoient par protocole Enhanced ShockBurst (protocole propriétaire RF de Nordic Semiconductor) à la centrale Nucleus. Hadès récupère les trames RF, les traitent puis publie toutes les données par WiFi (norme IEEE 802.11n) par protocole MQTT version 3.1.) sur le serveur MQTT E_BrokerServer. Ce serveur Broker MQTT est une Raspberry Pi externe à notre système. Il fonctionne grâce au logiciel E_BrokerMQTT, le logiciel employé est Moquitto version 1.4.14. Le client MQTT E_ClientMQTT est lancé notre StorageServer et est inscrit à notre serveur MQTT pour pouvoir recevoir ses données. Le client E_ClientMQTT fonctionne avec le logiciel MQTT.fx version 1.5.0. Notre StorageServer est une Raspberry Pi 3 Model B fonctionnant sous le système d'exploitation Raspbian Strech (version Septembre 2017). Il héberge une base de données sous SQLite version 3.20.1, elle est nommée DB. Le E_ClientMQTT envoie les données provenant du Broker MQTT dans la base de données DB, où elles sont stockées. Un utilisateur peut consulter ces données en lançant Poseidon depuis son E_Smartphone (Smartphone numérique Android fonctionnant sous la version Marshmallow 6.0, contenant l'application Poseidon développée Android Studio version 2.2.3). L'IHM de Poseidon est défini en partie **3.1.3**.

Pour communiquer avec ces différents périphériques la station, le smartphone et le ServerStorage devront utiliser et respecter des protocoles de communication spécifiques. Ces protocoles (désignés par le terme d'interface en UML) sont aussi présentés sur les différentes figures présentées.

Architecture Physique de Nucleus

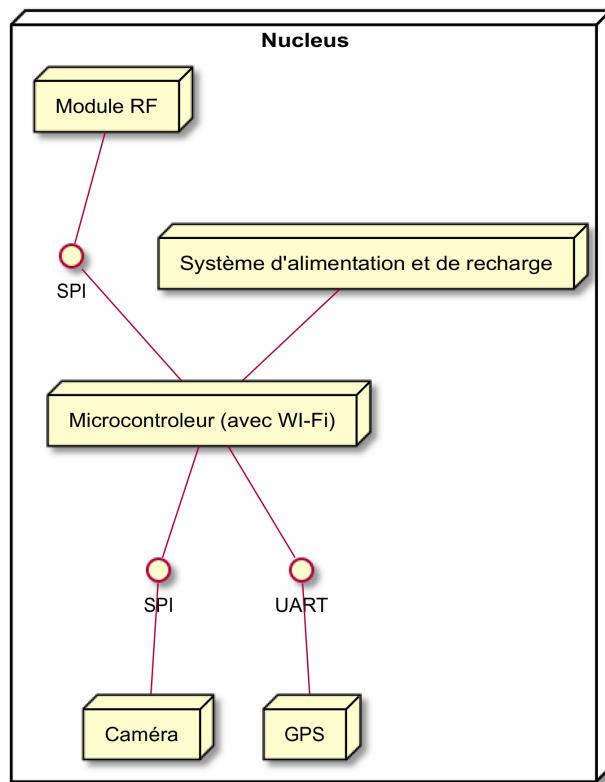


FIGURE 2.3 – Architecture matérielle de Nucleus

La centrale Nucleus est composé d'un microcontrôleur 32-bits avec un module Wi-Fi nommé ESP8266-01, d'un GPS NEO-6M GY-GPS6MV2 , d'une caméra ARDUCAM Mini-2Mpx OV2640, d'un module RF nRF24L01 ainsi que d'un système d'alimentation et de recharge (regroupant panneau solaire, batterie, circuit de recharge, etc. défini dans [CONC_MAT_PFE_VII]).

Architecture Physique de Thermos

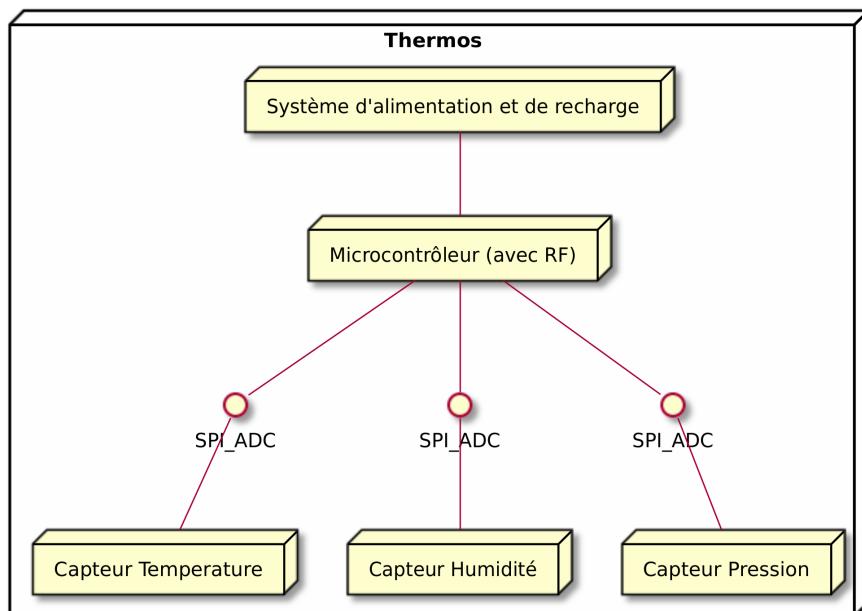


FIGURE 2.4 – Architecture matérielle de Thermos

Le noeud Thermos est composé d'un microcontrôleur 8-bits avec module RF nommé nRF24LE1, d'un capteur de température et d'humidité SHT21, d'un capteur de pression LPS25HBTR ainsi que d'un système d'alimentation et de recharge (regroupant panneau solaire, batterie, circuit de recharge, etc.). Tous ceci est défini dans [CONC_MAT_PFE_VI].

Architecture Physique de Luminos

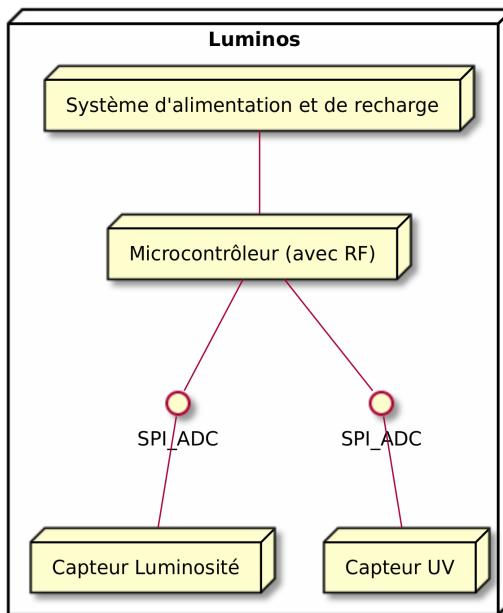


FIGURE 2.5 – Architecture matérielle de Luminos

Le noeud Luminos est composé d'un microcontrôleur 8-bits avec module RF nommé nRF24LE1, d'un capteur de luminosité OPT3001DNPT, d'un capteur UV ainsi que d'un système d'alimentation et de recharge (regroupant panneau solaire, batterie, circuit de recharge, etc.). Tous ceci est défini dans [CONC_MAT_PFE_VI].

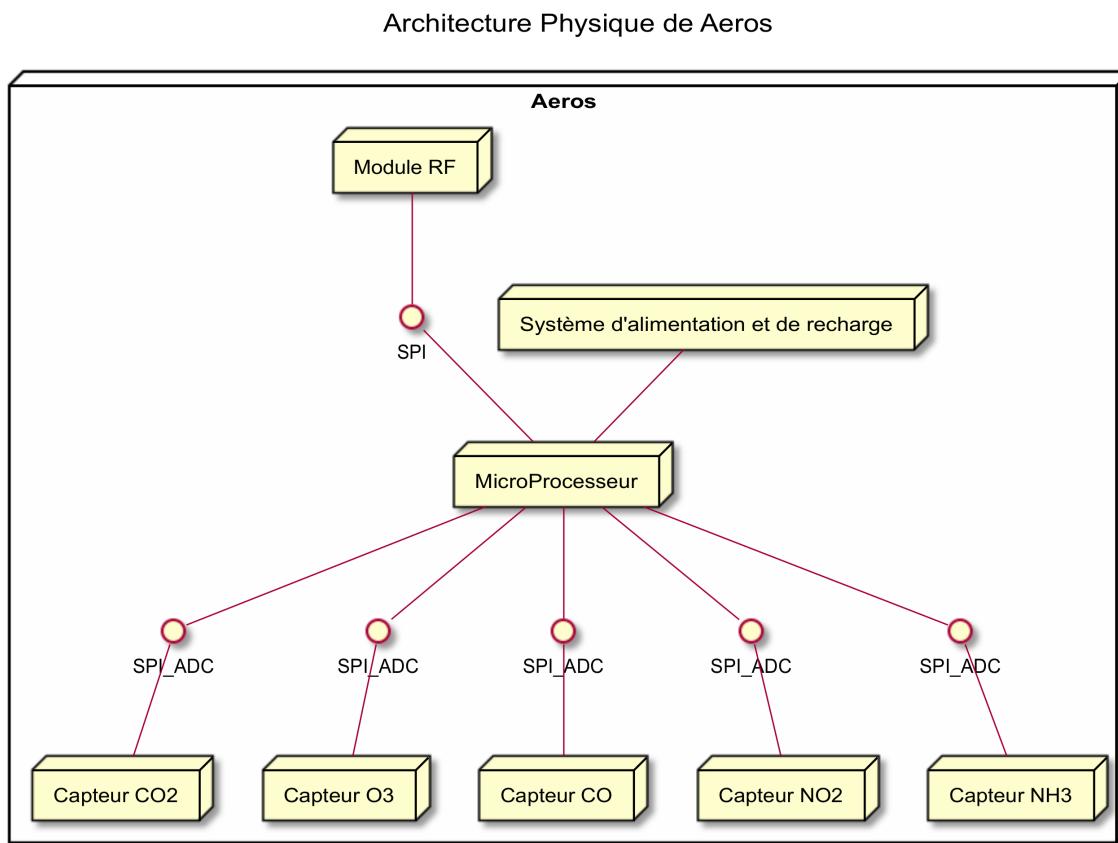


FIGURE 2.6 – Architecture matérielle de Aeros

Le noeud Aeros est composé d'un microcontrôleur 8-bits avec module RF nommé nRF24LE1, d'un capteur de dioxyde de carbone MG811, de capteurs de monoxyde de carbone MQ7, d'un capteur d'ozone MiCS 2614, d'un capteur MiCS 6814 mesurant le dioxyde d'azote, d'ammoniac et une mesure complémentaire de monoxyde de carbone, ainsi que d'un système d'alimentation et de recharge (regroupant panneau solaire, batterie, circuit de recharge, etc.). Tous ceci est défini dans [CONC_MAT_PFE_VI].

Architecture Physique de Precipitos

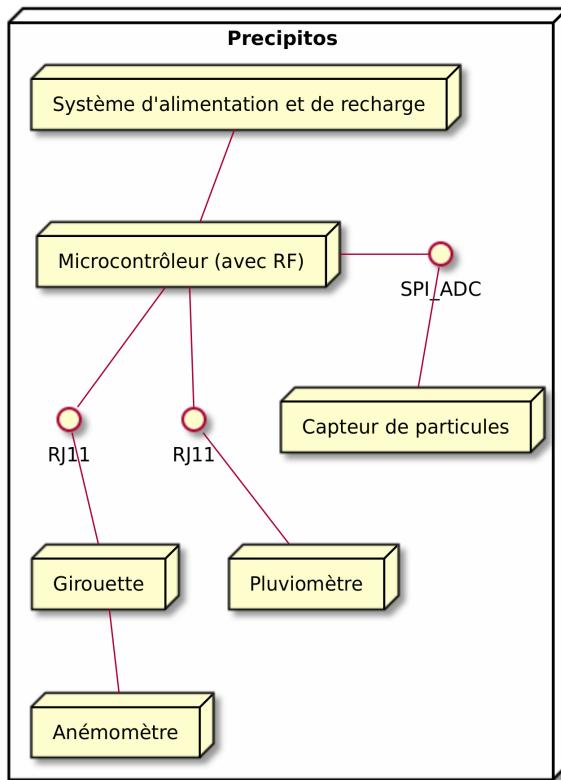


FIGURE 2.7 – Architecture matérielle de Precipitos

Le noeud Precipitos est composé d'un microcontrôleur 8-bits avec module RF nommé nRF24LE1, d'un capteur de particules (PM10) GP2Y1010AU0F, pluviomètre, d'une girouette, d'un anémomètre de la station météo WS3083 ainsi que d'un système d'alimentation et de recharge (regroupant panneau solaire, batterie, circuit de recharge, etc. défini dans [CONC_MAT_PFE_VI])

2.2.2 Interfaces du système

Cette partie décrit les entrées et sorties dites « logiques » et « physiques » du système. En effet, nous différencions dans cette étude deux grands types d'entrées/sorties :

- Celles dites de haut niveau (dites aussi logiques) qui décrivent les événements et données échangés entre l'utilisateur et les périphériques, ces entrées/sorties étant traitées ou pré-traitées par les périphériques avant d'être transmis au système. Ces entrées et sorties portent sur les intentions des acteurs interagissant l'application.
- Celles dites de bas-niveau (dites aussi physiques) qui sont les entrées/sorties réellement échangées entre le système et les périphériques.

2.2.2.1 Les interfaces logiques

La figure 2.8 présente le contexte en faisant figurer les entrées/sorties dites de haut niveau (ou logiques). Elles sont regroupées en grande famille. Pour représenter ce contexte logique, un diagramme de communication UML a été utilisé.

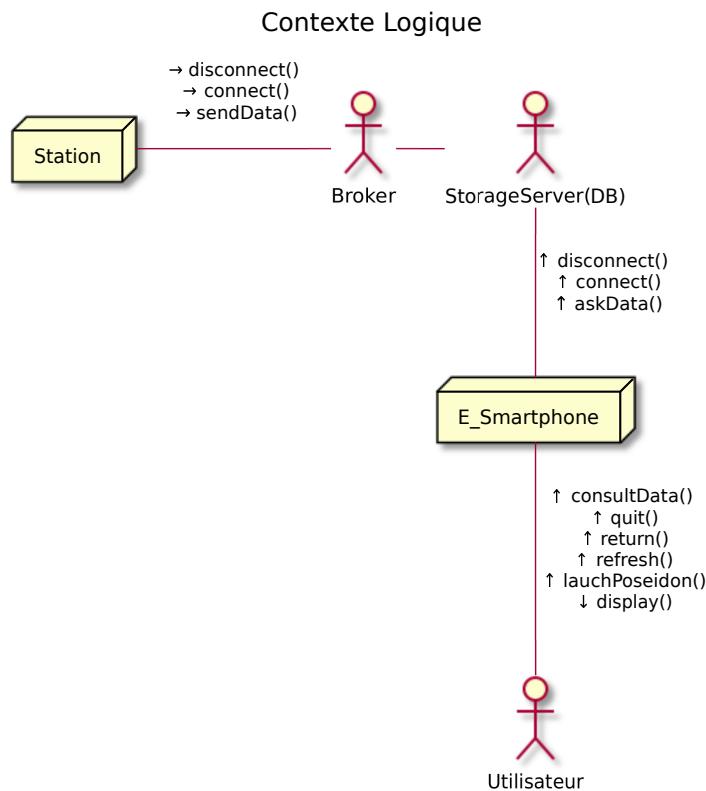


FIGURE 2.8 – Contexte logique

Dans ce diagramme de communication, seules les entrées/sorties logiques entre les acteurs et les périphériques du système sont présentées.

2.2.2.2 Les interfaces avec les acteurs

Les interactions entre l'utilisateur se font toutes via le E_Smartphone. Ces entrées et sorties logiques sont détaillées par la suite.

2.2.2.2.1 Entre l'utilisateur et le E_Smartphone

2.2.2.2.1.1 En provenance de l'utilisateur Voici les événements logiques en provenance de l'utilisateur vers le E_Smartphone :

- consultData() : l'utilisateur demande à consulter les données.
- launchPoseidon() : l'utilisateur demande à lancer l'application.
- quit() : l'utilisateur demande à quitter l'application.
- refresh() : l'utilisateur demande à rafraîchir la page.
- return() : l'utilisateur demande à revenir à l'écran précédent.

2.2.2.2.1.2 En provenance du E_Smartphone Voici les événements logiques en provenance de E_Smartphone vers l'utilisateur :

- display() : affiche les écrans et pop ups.

2.2.2.2.2 Entre le E_Smartphone et le ServerStorage

2.2.2.2.2.1 En provenance du E_Smartphone Voici les événements logiques en provenance de l'utilisateur vers le E_Smartphone :

- connect() : le E_Smartphone se connecte à la base de données.
- disconnect() : le E_Smartphone se déconnecte de la base de données.
- askData() : le E_Smartphone demande les données à la base de données.

2.2.2.2.2.2 En provenance de StorageServer Le StorageServer ne donne aucune directive à l'utilisateur. Il est passif et retourne les informations qu'on lui demande.

2.2.2.3 Entre la Station et le Broker

2.2.2.2.3.1 En provenance de la Station Voici les événements logiques en provenance de la Station vers le Broker :

- connect() : la Station se connecte au Broker.
- disconnect() : la Station se déconnecte du Broker.
- sendData() : la Station envoie les données au Broker.

2.2.2.2.3.2 En provenance du Broker Le Broker ne donne aucune directive à la Station. Il est passif et retourne les informations qu'on lui demande.

2.2.2.2.4 Entre le Broker et StorageServer L'ensemble des échanges entre le Broker et ServerStorage sont automatisés dans les logiciels de client et serveur MQTT, ils ne sont donc pas préciser dans cette specification. Ils sont bien-sûr unidirectionnel, du Broker vers StorageServer.

2.3 Fonctions principales développées

2.3.1 Rappel sur le cas d'usage

On assimile les cas d'usage aux étapes du cycle de vie d'un produit, chacun de ces cas d'usage définissant potentiellement plusieurs cas d'utilisation dissemblables. Le cycle de vie d'un produit commence par la fabrication, puis passe successivement par le conditionnement, la commercialisation, l'utilisation, la maintenance (en parallèle avec les autres cycles), la dés-installation et le recyclage.

2.3.2 Rappel sur le cas d'utilisation

Un cas d'utilisation (CU) est souvent lié à un cas d'usage, et représente un ensemble d'interactions entre au moins un acteur et le SaE. Les CU reposent sur deux types de représentations : une graphique, l'autre, textuelle. La première fait apparaître les acteurs sous forme de petits personnages et les cas d'utilisation dans des bulles, les deux étant liés par des flèches associatives.

Quant à la seconde, elle contient la description détaillée des fonctions employées et appelées du CU. Elles figurent dans différents scénarios dont le principal (le scénario nominal) fournit l'enchaînement logique d'une utilisation représentative du SaE.

Titre	Rappelle en quelques mots l'objectif principal du cas d'utilisation.
Résumé	Décrit brièvement le comportement du cas d'utilisation.
Portée	Définit la portée de conception du CU (étendue spatiale).
Niveau	Niveau de granularité du cas d'utilisation (Stratégique, utilisateur ou sous-fonction).
Acteur principal	Acteur qui déclenche le CU.
Acteurs secondaires	Acteurs qui participent au CU.
Pré-conditions	Ensemble des conditions qui doivent être vérifiées avant le déroulement du CU. Les pré-conditions, sans mention contraire explicite, des CU parents au CU doivent toujours être vérifiées.
Garanties minimales	Définit ce qui est garanti par le système à l'étude même en cas d'échec du cas d'utilisation.
Garanties en cas de succès	Définit les garanties en cas de succès (par le scénario nominal ou par ses variantes).
Scénario nominal	C'est un scénario représentatif de l'utilisation du système où tout se passe bien. Il se termine par la réussite des objectifs. Il est constitué d'une condition déclenchant le scénario, d'un ensemble d'étapes, d'une condition de fin, et éventuellement d'extensions ou de variantes. Une étape peut être une interaction entre acteurs, une étape de validation, ou un changement interne.

Variantes	Ce sont des scénarios représentatifs de l'utilisation du système où tout se passe bien. Ils se terminent par la réussite des objectifs. Ils sont constitués d'une condition déclenchant le scénario, d'un ensemble d'étapes, d'une condition de fin, et éventuellement d'extensions ou de variantes. Une étape peut être une interaction entre acteurs, une étape de validation, ou un changement interne.
Extensions	Définissent les autres scénarios que le scénario nominal (par exemple ceux qui se terminent par un échec). Ils se déclenchent sur des conditions spécifiques détectées par le SaE.
Informations complémentaires	Informations diverses nécessaires à la compréhension du CU

2.3.3 CU Informer périodiquement les utilisateurs afin d'anticiper les risques de réactions allergiques

2.3.3.1 Description graphique

Diagramme de Cas d'Utilisation Principal

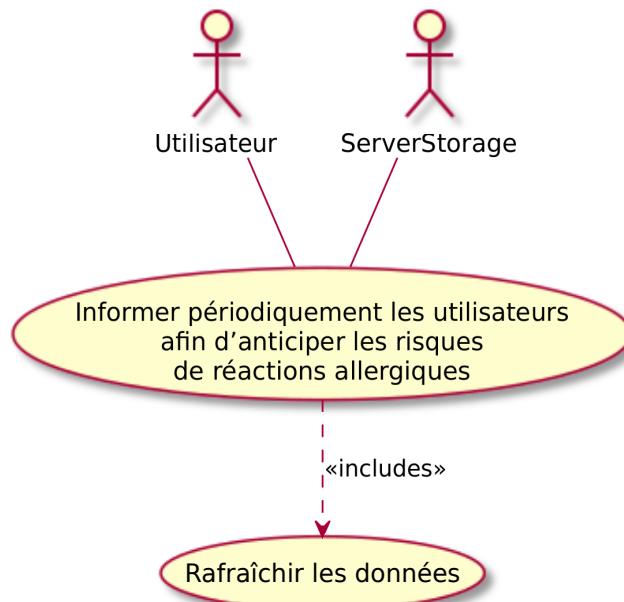


FIGURE 2.9 – CU Principal

2.3.3.2 Description textuelle

Titre	Informer périodiquement les utilisateurs afin d'anticiper les risques de réactions allergiques.
Résumé	L'utilisateur souhaite être informé d'une éventuelle vague de pollen. En lançant l'application Android, il visualise les données et vérifie s'il existe une anomalie.
Acteur principal	L'utilisateur.
Acteur secondaire	ServerStorage(DB).
Pré-conditions	La valise est alimentée, la tablette est suffisamment chargée. L'ensemble du système est prêt et fonctionnel.
Garanties minimales	-
Garanties en cas de succès	L'utilisateur a visualisé les données météorologiques.

Scénario nomi-nal	<ol style="list-style-type: none"> 1. L'utilisateur lance Poseidon. 2. L'utilisateur affiche Screen>Loading. 3. Poseidon <u>rafraîchit les données</u>. 4. Poseidon affiche Screen>Home. 5. L'utilisateur demande de visualiser les données sur Poseidon. 6. Poseidon affiche Screen>Data. 7. L'utilisateur demande de retourner à l'accueil. 8. Poseidon affiche Screen>Home. 9. L'utilisateur demande de quitter Poseidon. 10. Poseidon se ferme.
Variantes	<p>5.a :[L'utilisateur demande de quitter Poseidon] 5.a.1. Va en 9.</p> <p>9.a :[L'utilisateur demande de visualiser les données à nouveau] 9.a.1. Retour en 5.</p>
Extensions	<p>4-8.a : L'utilisateur n'a pas quitté Poseidon 4-8.a.1. Poseidon <u>rafraîchit les données toutes les T_DR.</u></p>

TABLE 2.2 – Tableau - CU Principal

2.3.4 CU Rafraîchir les données

2.3.4.1 Description graphique

N.A.

2.3.4.2 Description textuelle

Titre	Rafraîchir les données.
Résumé	Poseidon démarre une connexion, récupère les données et se déconnecte de DB.
Acteur principal	-
Acteur secondaire	ServerStorage(DB)
Pré-conditions	-
Garanties minimales	-
Garanties en cas de succès	L'utilisateur a visualisé les données météorologiques.
Scénario nominal	<ol style="list-style-type: none"> 1. Poseidon demande à se connecter à ServerStorage(DB). 2. ServerStorage(DB) accepte la connexion. 3. Poseidon demande les données les plus récentes des capteurs. 4. ServerStorage(DB) envoie les données à Poseidon. 5. Poseidon se déconnecte de ServerStorage(DB).
Variantes	<p>5.a :[L'utilisateur est présent sur l'écran Screen_Loading]</p> <p>5.a.1. Poseidon se déconnecte de DB. 5.a.2. Poseidon attend T_LD.</p> <p>5.a :[L'utilisateur est présent sur l'écran Screen_Data]</p> <p>5.a.1. Poseidon se déconnecte de DB. 5.a.2. Poseidon rafraîchit son écran Screen_Data.</p>
Extensions	<p>2 :</p> <p>2.a :[La connexion est refusé - indisponible]</p> <p>2.a.1. DB refuse la connexion. 2.a.2. Poseidon affiche PopUp_ConnectionError.</p> <p>2.b :[Poseidon n'est pas connecté au réseau]</p> <p>2.b.1. Poseidon affiche PopUp_ConnectionError.</p>

TABLE 2.3 – Tableau - CU Rafraîchir les données des capteurs

2.4 Contraintes

2.4.1 Politique réglementaire

N.A

2.4.2 Contraintes matérielles

Pour le développement de la station de qualité de l'air, nous devons prendre en compte la gestion de l'énergie des différents modules et leur autonomie. La caméra doit prendre des photos de qualité supérieur ou égale à 2 MP. Le placement des différents noeuds possède des contraintes, elles doivent être respecté lors de l'installation du système. La procédure est détaillée dans le manuel d'installation [Manuel d'Installation].

2.4.3 Contraintes environnementales

Les différents capteurs de la station se doivent d'être appropriés à l'utilisation en extérieur. Il faut prendre en compte les perturbations de l'environnement dont la disponibilité de la bande de fréquence et les interférences.

2.4.4 Exigences de fiabilité

Les mesures récupérés par la station se doivent d'être correctes afin de permettre une analyse des données qualitative.

2.4.5 Exigences de maintenabilité

La maintenance de la station et de ses logiciels est à la charge des personnes mettant en place le système. Après livraison notre équipe n'interviendra plus sur la station.

2.4.6 Exigences de disponibilité

La station est présent dans la ville d'Angers. L'application possédant les données traitées sera à la disposition des habitants de l'agglomération. La station doit être opérationnelle sans discontinuité.

2.5 Hypothèses et dépendances

Ce paragraphe décrit les hypothèses minimalistes prétendues vraies dans la suite du dossier.

- La centrale est connectée au réseau Wi-Fi de la ville.
- Les nœuds de capteurs sont reliés à la centrale par protocole Enhanced Shockburst.
- Version Android supérieure ou égale à la version 6.0
- Poseidon est adapté pour un écran de smartphone ou tablette tactile (Base : Large et X-Large).
- Prise en charge de l'orientation portrait et paysage sur l'appareil hôte de Poseidon.

Chapitre 3

EXIGENCES SPÉCIFIQUES

3.1 Interface Homme - Machine

3.1.1 Généralités

La plateforme de récupération des données sera exclusivement utilisable via l'application Android.

L'utilisateur pourra depuis cette application visualiser les données récoltées par la station.

3.1.2 Action des utilisateurs

3.1.2.1 Diagramme d'enchaînement des vues de Poseidon

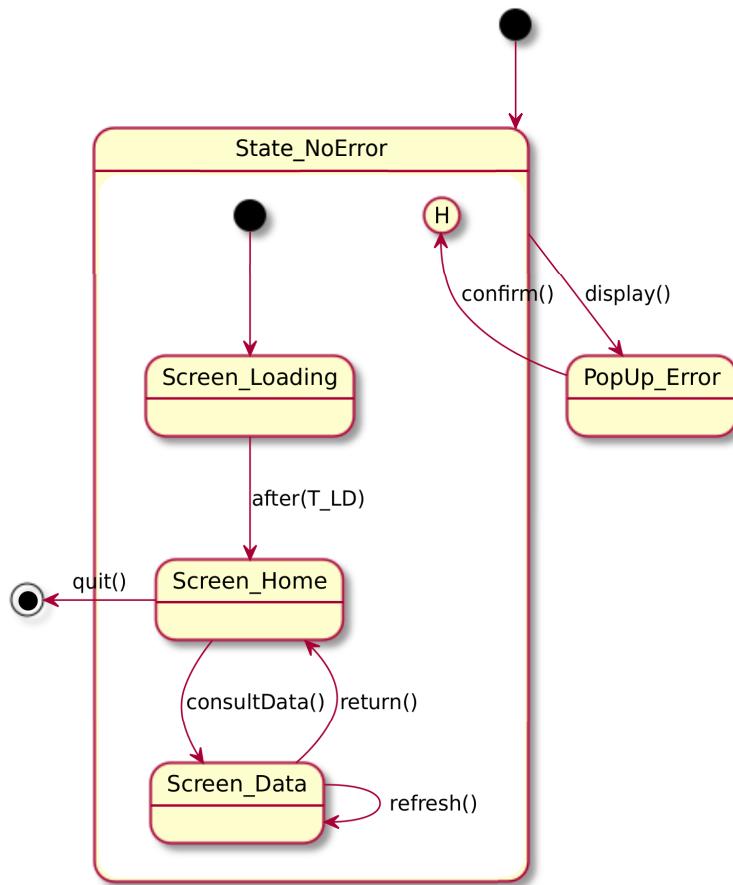


FIGURE 3.1 – Diagramme d'enchaînement des vues de Poseidon

permet

3.1.3 Les écrans

Dans la partie suivante, nous présenterons la navigation possible entre les différents écrans de l'IHM de l'application Poseidon. Il y aura également une précision sur les actions possibles sur chaque écran.

Ici la question de l'UX/UI (User Experience/User Interface) était simple dans le cadre où l'application est avant tout destinée récupérer les informations, celles-ci ne seront traitées que plus tard.

L'utilisateur pourra sur chaque écran exécuter une instruction de retour par une pression sur le bouton prévu à cet effet sur l'appareil ou sur le système Android.

L'équipe de réalisation se réserve le droit d'exercer des modifications sur les écrans : Couleurs, Positionnement, Forme, Police, Taille de la police. Ces images sont donc à titre indicatif.

3.1.3.1 Écran de chargemennt

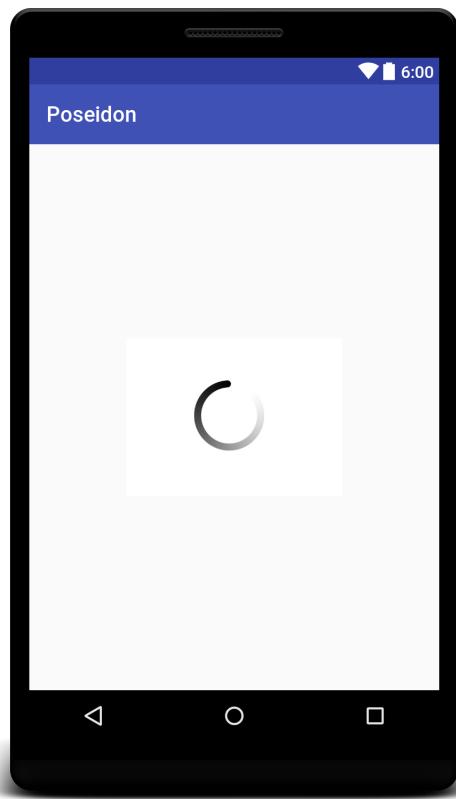


FIGURE 3.2 – Vue de Screen_Loading

L'écran Screen_Loading apparaît lors du lancement de l'application. Il reste affiché pendant la durée T_LD. Il laisse le temps à l'application de charger les données depuis DB.

3.1.3.2 Écran d'accueil

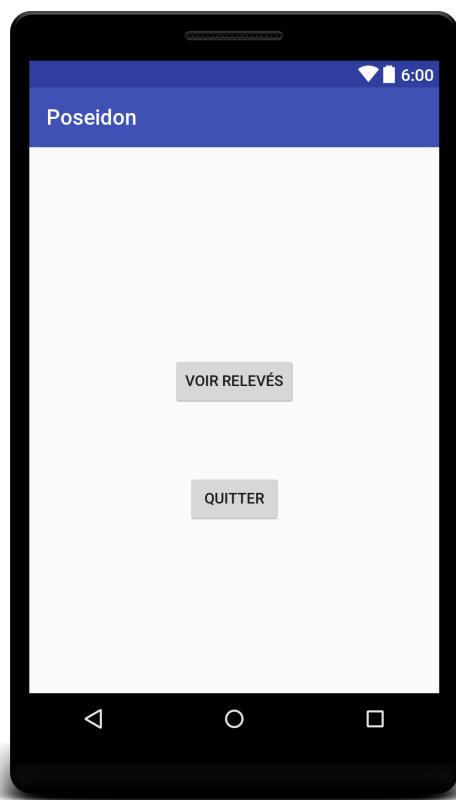


FIGURE 3.3 – Vue de Screen_Home

L'écran principal Screen_Home est l'écran qui apparaît après lancement de l'application. Celui-ci ne contient pour le moment que d'un de 2 choix : Voir les relever qui permet d'arriver sur un second d'écran de données ou Quitter.

3.1.3.3 Ecran de visualisation des données

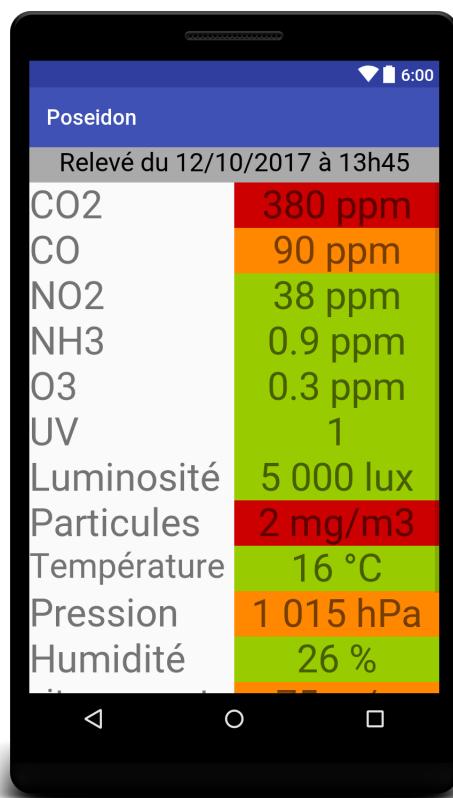


FIGURE 3.4 – Vue de Screen_Data (1/2)

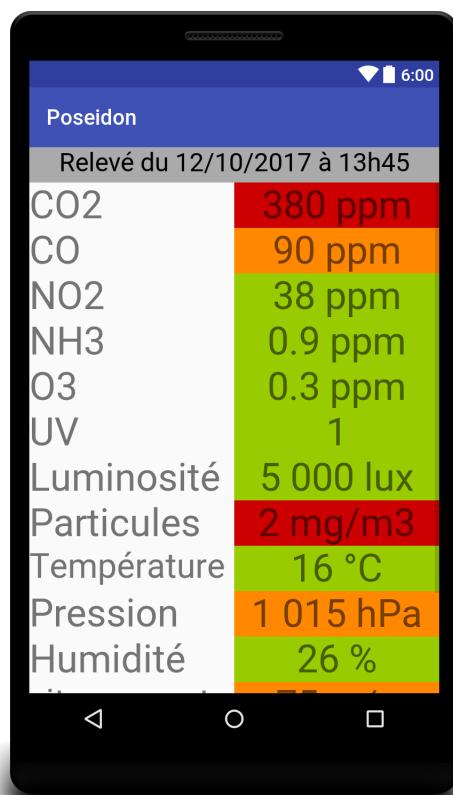


FIGURE 3.5 – Vue de Screen_Data (2/2)

L'écran Screen_Data est accessible depuis le Screen_Home. Il affiche simplement les données telles que stockées dans la base de données. Les données sont accompagnées de leur unité dans le système international ainsi que d'une couleur indiquant si la valeur mesuré est dangereuse pour l'homme ou non. L'indicateur vert signifie qu'il y a pas de danger, rouge signifie que la mesure est dangereusement élevé pour l'homme, orange signale une mesure médiane. La date et l'heure des mesures est affiché en haut de l'écran.

3.1.3.4 Pop Up erreur de connexion

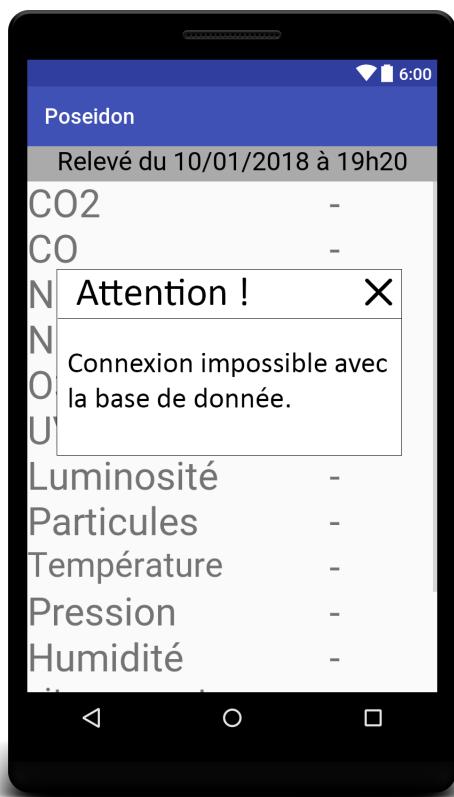


FIGURE 3.6 – Vue de PopUp_ConnectionError

La PopUp_ConnectionError apparaît à l'écran lorsque le rafraîchissement des données ne peut pas être fait, dû à un problème de connexion entre Poseidon et DB.

3.2 Dictionnaire du domaine

Aeros : Module spécifique aux capteurs d'air tels que les gaz (CO, CO₂, NO₂, O₃ ...) et les poussières de diamètre comparable aux grains de pollen.

Apollon : Logiciel embarqué sur Luminos qui récupère périodiquement des informations de capteurs.

Athena : Logiciel embarqué sur Thermos qui récupère périodiquement des informations de capteurs.

Broker : Serveur externe au système, qui permet de retransmettre de l'information. Aussi appelé E_BrokerMQTT.

DB : La base de données mise en place dans le StorageServer.

E_Smartphone : Smartphone numérique Android fonctionnant sous la version Marshmallow 6.0, contenant l'application MasterMind développée sous Android studio version 2.2.3.

Ethernet : Protocole de réseau local à commutation de paquets. C'est une norme internationale : ISO/IEC 8802-3.

GPIO : Port d'entrées et de sorties du PDU.

Hades : Logiciel embarqué sur Nucleus qui récupère périodiquement des informations de GPS et une image >= 2 MP.

Hermes : Logiciel embarqué sur Aeros qui récupère périodiquement des informations de capteurs.

IP : (Internet Protocol) numéro d'identification de tout réseau pour une connexion internet.

Luminos : Module spécifique aux capteurs de lumière et d'UV.

Module : Elément unitaire qui fait partie d'un ensemble.

Noeud : Elément unitaire d'un réseau, fait référence à un noeud de capteurs composé de plusieurs capteurs d'un microprocesseur et communique avec une station.

Nucleus : Une centrale d'acquisition autonome en énergie, capable de collecter et transmettre par WiFi à un serveur privé ou public les données issues de différents capteurs.

Poseidon : Logiciel Android s'exécutant sur un smartphone ou une tablette.

Précipitos : Module spécifique aux capteurs de précipitations tel que le pluviomètre, l'anémomètre et la girouette.

StorageServer : La carte Raspberry herbegeant DB et le client MQTT.

Station : Système représentant l'ensemble de la Nucleus et des noeuds de capteurs.

T_DR : (Time of Data Refreshing) Temps fixe auquel on rafraîchit les données de Poséidon par rapport à DB. (Evenement)

T_LD : (Time of Loading Data) Temps fixe pendant lequel on attend le chargement des données sur l'application.

Thermos : Module spécifique aux capteurs de pression, d'humidité et de température.

Zeus : Logiciel embarqué sur Précipitos qui récupère périodiquement des informations de capteurs.

Table des figures

2.1	Architecture matérielle et logicielle	11
2.2	Légende du diagramme de déploiement UML	12
2.3	Architecture matérielle de Nucleus	13
2.4	Architecture matérielle de Thermos	13
2.5	Architecture matérielle de Luminos	14
2.6	Architecture matérielle de Aeros	15
2.7	Architecture matérielle de Precipitos	16
2.8	Contexte logique	17
2.9	CU Principal	21
3.1	Diagramme d'enchaînement des vues de Poseidon	27
3.2	Vue de Screen_Loading	28
3.3	Vue de Screen_Home	29
3.4	Vue de Screen_Data (1/2)	30
3.5	Vue de Screen_Data (2/2)	30
3.6	Vue de PopUp_ConnectionError	31

Liste des tableaux

1.1	Tableau des acronymes et abréviations	7
1.2	Tableau des références	8
2.2	Tableau - CU Principal	22
2.3	Tableau - CU Rafraîchir les données des capteurs	23

Chapitre 4

VALIDATION DU DOCUMENT

Document fait à Angers, le 5 Octobre 2017.

Pour l'équipe de réalisation,
Paraphes, "Lu et approuvé"

M. KOUATLI Sami,
Prestataire, Membre du projet