



RÉGION ACADÉMIQUE
PAYS DE LA LOIRE

*Liberté
Égalité
Fraternité*



Lycée
**TOUCHARD
WASHINGTON**

Dossier d'analyse

BTS CIEL SESSION 2025 - IR E6 – PROJET TECHNIQUE

Système de Surveillance météorologique connecté



Lycée : Touchard Washington		Ville : LE MANS
Nom du projet :	Système de surveillance météorologique connecté	
N° du projet :	TW3	
Projet nouveau	Oui <input type="checkbox"/> Non <input checked="" type="checkbox"/>	Projet interne : Oui <input checked="" type="checkbox"/> Non <input type="checkbox"/>
Délai de réalisation	150 heures	Statut des étudiants : Formation initiale <input checked="" type="checkbox"/> Apprentissage <input type="checkbox"/>
Spécialité des étudiants	ER <input type="checkbox"/> IR <input checked="" type="checkbox"/> Mixte <input type="checkbox"/>	Nombre d'étudiants : 3 étudiants
Professeurs responsables	Philippe CRUCHET , François MARTIN, Anthony LE CREN, Jilali KHAMLACH, Saïd LAHSIKA.	

Sommaire

Introduction.....	3
Objectifs.....	4
Répartition des tâches.....	5
Descriptions des capteurs et des actionneurs.....	6
Description des technologies.....	7
Entrées physique des capteurs.....	9
Dictionnaire de données.....	9
Trames reçues et envoyées.....	10
Diagramme BPMN du récepteur et des clients Android.....	12
Diagramme BPMN du client web.....	14
Dictionnaire des acteurs.....	15
Description des cas d'utilisation.....	16
Traiter les données en provenance des capteurs.....	17
Diagramme de séquence.....	19
Stocker les données météorologiques.....	20
Diagramme de séquence.....	21
Diffuser les données météorologiques en temps réel.....	23
Diagramme de séquence.....	25
Configurer les paramètres météorologique.....	26
Afficher les données météorologiques en temps réel.....	29
Diagramme de séquence.....	30
Afficher les données historiques.....	32
S'authentifier.....	34
Diagramme de séquence.....	36
Matrice RACI.....	37
Récepteur Météorologique.....	38
Contrôleur Météorologique.....	39
Client Android.....	41
IHM.....	42
Client Web.....	42
.....	42
.....	43
Client Android.....	44
Menu principal.....	44
Menu température.....	44
Menu capteurs.....	45
Menu paramètres généraux.....	45
Menu réglages des capteurs.....	46
Menu configuration des seuils.....	46
Pluviogramme.....	47
Diagramme de flux de données.....	48
Description de la base de données.....	49
Gantt.....	51

Introduction

Le projet consiste à développer un système de surveillance météorologique connecté, basé sur une infrastructure réseau et un ensemble de capteurs pour collecter et afficher des données climatiques.

Son objectif est de permettre un suivi en temps réel des conditions météorologiques locales à distance, de réaliser un stockage de ces données pour une analyse historique. Les domaines d'application sont nombreux, agriculture, surveillance environnementale, ou même pour les particuliers désireux de connaître précisément la météo.



Image 1: Synoptique

Comme présenté ci-dessus, le système météorologique est centré autour d'un serveur Raspberry, qui réceptionnera les données des capteurs, les traitera afin de les stocker et de les diffuser en temps réel. Pour la réception des données, cela se fera à partir d'une clé RTL-DSR et de l'application RTL_433 qui permettra la traduction des trames. Les données seront donc stockées au sein de la Raspberry qui fera base de données. Les données seront diffusées grâce à une box Internet. Les clients pourront se connecter à partir d'un navigateur web(en local et à distance) ou d'un appareil Android. Les clients recevront les données en temps réel, alors que les clients web auront accès aux données présentes dans la base de données.

Objectifs

Aspect	Détail
Objectif du projet	Concevoir et déployer un système de surveillance météorologique connecté.
Fonctionnalités principales	<div><div>- Collecter des données climatiques en temps réel</div><div>- Stocker les données</div><div>- Diffuser les données localement et à distance</div></div>
Données météorologiques collectées	<div><div>- Température</div><div>- Humidité</div><div>- Vitesse et direction du vent</div><div>- Précipitations</div><div>- Pression atmosphérique</div></div>
Portée géographique	Collecte locale et diffusion des données à distance.

Répartition des tâches

Étudiants	Fonctions à développer et tâches à effectuer	
Étudiants 1: Bertault Nolhan ER IR	Liste des fonctions assurées par l'étudiant <ul style="list-style-type: none"> ➤ Traiter les données en provenance des capteurs. ➤ Stocker les données météorologiques. ➤ Diffuser les données météorologiques en temps réel. 	Installation : Raspi OS sur le Raspberry Pi Serveur de base de données Mise en œuvre : Logiciel pour collecter les données météorologiques RTL_433 Configuration : Filtrage des sources de données réceptionnées par la clé RTL-SDR. Réalisation : Fonctions en charge Documentation : Contribution au dossier commun, rédaction de sa partie individuelle.
Étudiants 2: Deneau Maverick ER IR	Liste des fonctions assurées par l'étudiant <ul style="list-style-type: none"> ➤ Afficher les données historiques. ➤ S'authentifier. 	Installation : Serveur Web Mise en œuvre : Highcharts, Datatables Configuration : Serveur Web, accès sécurisé au site Web à partir de la box Fortinet Réalisation : Fonctions en charge Documentation : Contribution au dossier commun, rédaction de sa partie individuelle.
Étudiants 3: Bouhaigour Marwan ER IR	Liste des fonctions assurées par l'étudiant <ul style="list-style-type: none"> ➤ Afficher les données météorologiques en temps réel. ➤ Configurer les seuils d'alerte météorologique. 	Installation : Box Fortinet et câblage du réseau Mise en œuvre : Programmation QML Configuration : Accès Wifi sur la Box Fortinet Réalisation : Fonctions en charge Documentation : Contribution au dossier commun, rédaction de sa partie individuelle.

* Chaque tâche spécifique à un étudiant sera donc indiquée grâce aux couleurs présentent dans le tableau ci-dessus. De plus en pied de page le nom de l'étudiant l'ayant écrit est indiqué, si aucun nom n'est indiqué cela signifie que plusieurs élèves ont écrit la page.

Descriptions des capteurs et des actionneurs

Hygro/Thermomètre	
Référence	Nexus-TH
Rôle	Déterminer la température et le taux d'humidité
Caractéristique	<ul style="list-style-type: none"> ➤ Température: -40°C à 50°C: de $\pm 1^\circ\text{C}$ à $\pm 4^\circ\text{C}$ ➤ Humidité: 20% à 95%: de $\pm 5\%$ à $\pm 10\%$

Anémomètre	
Référence	WS-2300-15
Rôle	Déterminer la vitesse et la direction du vent
Caractéristique	Vitesse: <ul style="list-style-type: none"> ➤ Maximum: 180km/h ou 50m/s ➤ Précision: 0.1m/s Direction: <ul style="list-style-type: none"> ➤ Résolution graphique de 22.5°

Pluviomètre	
Référence	WS-2300-16
Rôle	Déterminer le niveau de précipitation
Caractéristique	<ul style="list-style-type: none"> ➤ Durée: 1 à 24h ➤ Maximum : 999.9mm ➤ Précision: 0.1mm

Capteur Local	
Référence	BME280
Rôle	Déterminer la pression atmosphérique, la température et l'humidité
Caractéristique	<ul style="list-style-type: none"> ➤ Température: -40°C à +85°C: $\pm 1^\circ\text{C}$ ➤ Humidité: 0% à 100%: $\pm 3\%$ ➤ Pression Atmosphérique: 300hPa à 1100hPa: $\pm 1\text{Pa}$

Description des technologies

➤ Capteur météorologique :

- Doit couvrir les principaux paramètres climatiques mentionnés.
- Alimentation autonome recommandée pour une installation en extérieur.

➤ Récepteur météo avec clé RTL-SDR :

- Capable de capter des signaux radio HF 433 MHz. Dans un rayon de 50 mètres.
- Connectivité USB pour la transmission des données vers le Raspberry Pi.

➤ Serveur Raspberry Pi :

- Stockage suffisant pour l'historique des données (prévoir une capacité d'extension via un disque dur externe).
- Système d'exploitation Raspi OS ou équivalent, avec logiciel de traitement des données en temps réel.
- Alimentation sauvegardée afin de recevoir les données même en

cas de coupure de courant.

➤ Box Internet :

- Sécurise l'accès au Raspberry Pi.
- Gère le réseau Wi-Fi et Ethernet pour les différents appareils.

➤ Clients Web et Android :

- Le client Web doit être compatible avec les principaux navigateurs (Chrome, Firefox, etc.) il permet de visualiser les données historiques.
- L'application Android doit être légère, avec des affichages en temps réel.

Entrées physique des capteurs

Grandeur physique	Unité	Échelle de mesure	Précision	Type de grandeur	Capteur
Température	°C	-40 à 65	1°C	Analogique	Hygro/Thermomètre
Humidité	%	1 à 99	5%	Analogique	
Position angulaire	degré	0 à 360	22,5°	Analogique	Anémomètre
Vitesse	km/h	0 à 180	0,1 m/s	Analogique	
Précipitation	mm	0 à 999,9	0,1 mm	Numérique	Pluviomètre
Baromètre	hPa	300 à 1100	1 Pa	Analogique	Capteur Local

Dictionnaire de données

Nom	Type	Taille/Précision	Exemple
Température	réel	dixième de degré	22.5
Humidité	entier	2 chiffres	23
Vitesse du vent	réel	dixième de m/s	7.344
Direction du vent	réel	22.5°	225
Précipitations	réel	dixième de mm	136.800
Pression Atmosphérique	entier	4 chiffres	82.68

Trames reçues et envoyées

Le système utilise des trames JSON, le logiciel RTL_433 peut créer ces trames de manière autonome ce qui permet un traitement simplifié des données. La structure d'une trame se présente comme ci-dessous.

{<Date>, <Marque>, <Modèle>, <Identifiant>, <Canal>, <Batterie> [,<Données>, <Données>]}

Champs (peuvent changer en fonction de la trame)	Nom	Exemple de valeur
Date (Obligatoire)	time	"2020-02-12 16:14:09"
Marque (Obligatoire)	brand	"OS"
Modèle (Obligatoire)	model	"Fineoffset-WHx080"
Identifiant du capteur (Obligatoire)	id	169
Canal (Obligatoire)	channel	1
Batterie (Obligatoire)	battery	1
Température	temperature_C	21.400
Humidité	humidity	37

Exemple : {"time" : "2020-02-12 16:14:09", "brand" : "OS", "model" : "Oregon-THGR810", "id" : 169, "channel" : 1, "battery_ok" : 1, "temperature_C" : 21.400, "humidity" : 37}

Lors de la diffusion de données nous allons aussi utiliser le format JSON. Afin de faciliter l'écriture et la lecture des données, nous allons réduire le nombre d'informations dans une trame, car certaines sont superflues. La nouvelle structure est représentée ci-dessous.

{Date,[<Données>,<Données>]}

Champs (peu changer en fonction de la trame)	Nom	Exemple de valeur
Date (Obligatoire)	date	"2020-02-12 16:14:09"
Température	temperature	21.400
Humidité	humidite	37
Vitesse du vent	vitesse-vent	7.344
Direction du vent	direction-vent	225
Pression	pression	35
Précipitation	precipitation	136.800

Exemple : {"date" : "2020-02-12 16:14:09", "temperature" : 21.400, "humidite" : 37, "vitesse-vent" : 7.344, "direction-vent" : 225, "pression" : 35, "precipitation" : 136.800}

Diagramme BPMN du récepteur et des clients Android

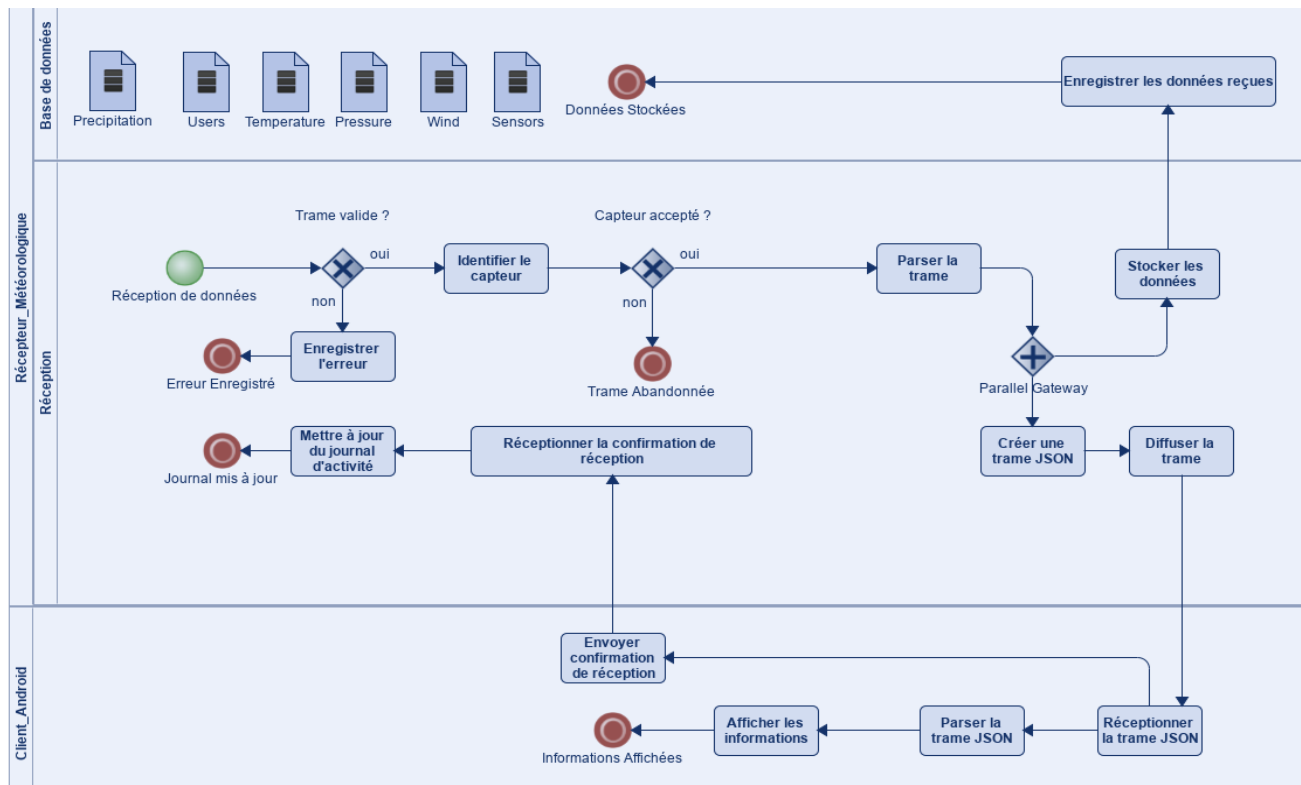


Image 2: Diagramme BPMN "Récepteur et Client Android"

Ce diagramme BPMN (Business Process Model and Notation) décrit le processus métier impliqué dans la réception, le traitement, le stockage et la diffusion des données météorologiques au sein d'un système. Le processus met en relation deux acteurs principaux :

1. **Le récepteur de données météo** : Il collecte des informations climatiques en temps réel, telles que la température, l'humidité, la direction et la vitesse du vent, ainsi que les précipitations. Ces données sont ensuite traitées et enregistrées dans une base de données structurée.
2. **Le client Android** : Une application cliente qui reçoit les données sous forme de trame JSON, les interprète et les affiche à l'utilisateur. Le client Android envoie également un accusé de réception, qui est enregistré dans un journal pour assurer la traçabilité et la fiabilité des échanges.

Le processus détaillé comprend les étapes suivantes :

- **Réception et traitement des données** : Les données météorologiques sont reçues, analysées et transformées en un format standardisé prêt à être stocké.
- **Stockage dans la base de données** : Les informations traitées sont sauvegardées dans une table de données dédiée.
- **Génération et diffusion de la trame JSON** : Une trame JSON contenant les données météo est générée et envoyée au client Android.
- **Traitement par le client Android** : Le client Android décode et affiche les informations pour l'utilisateur final, tout en envoyant un accusé de réception au récepteur météo.

Diagramme BPMN du client web

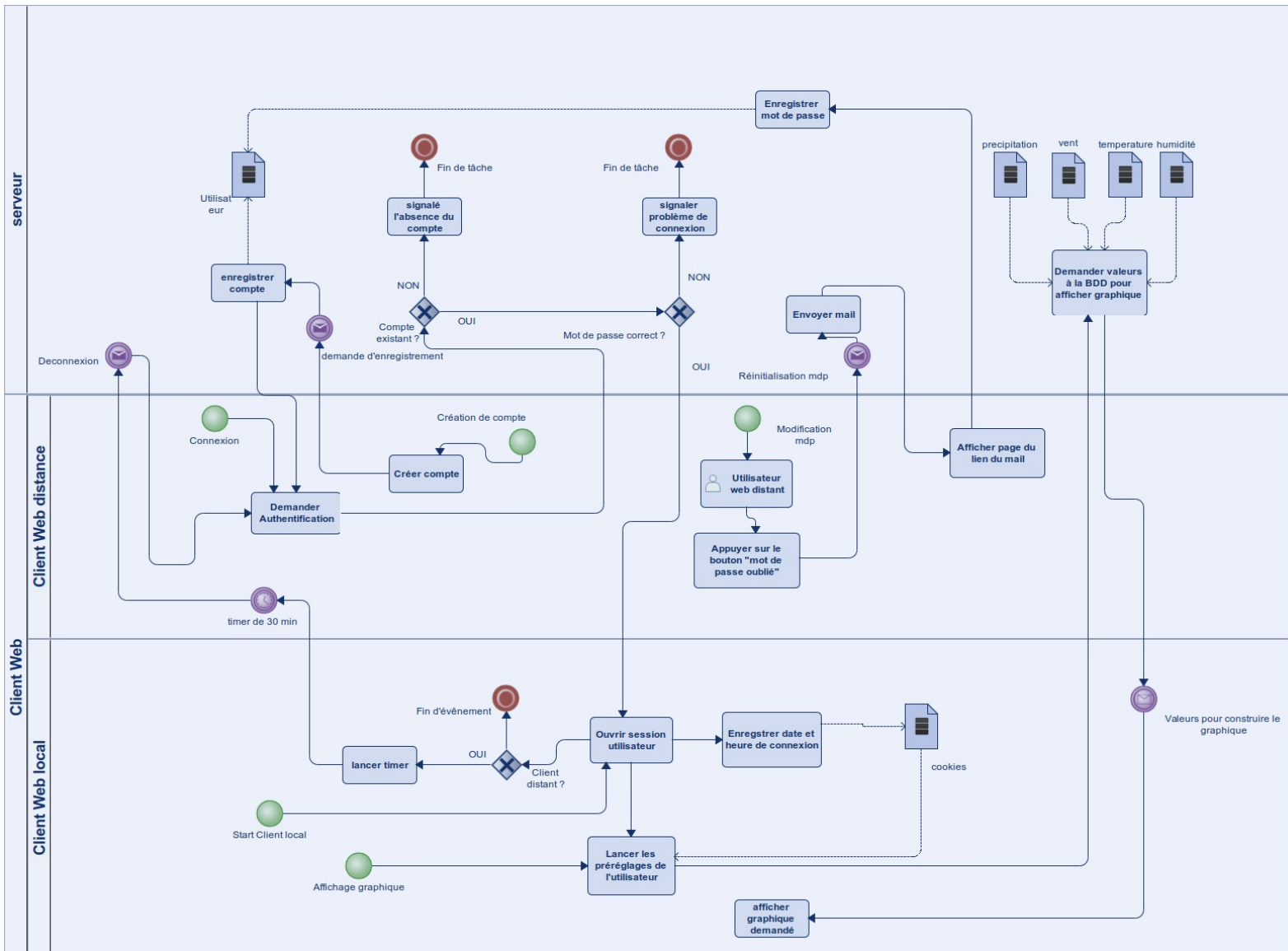


Image 3: Diagramme BPMN "client web"

Ce diagramme est un diagramme de processus BPMN (Business Process Model and Notation) qui représente le fonctionnement d'un système de gestion de compte et d'affichage de données météorologiques. Il est divisé en plusieurs pools et lanes correspondant aux différents acteurs et systèmes impliqués.

Structure du diagramme :

❖ **Pools et Lanes :**

- **Serveur** : Il gère l'enregistrement des comptes, l'authentification, la récupération de mot de passe et l'affichage des données.
- **Client Web distant** : Permet la connexion/déconnexion des utilisateurs et l'interaction avec le serveur.
- **Client Web local** : S'occupe du lancement du client, de l'affichage graphique et de la gestion des cookies.

❖ **Principaux processus représentés :**

- **Création de compte** : L'utilisateur enregistre son compte. Si le compte existe déjà, il passe par l'authentification. Sinon, un nouveau compte est créé.
- **Connexion et authentification** : Un utilisateur tente de se connecter. Si le mot de passe est incorrect, une erreur est signalée. Sinon, la session utilisateur s'ouvre et les préférences sont chargées.
- **Réinitialisation de mot de passe** : L'utilisateur appuie sur "mot de passe oublié". Un mail est envoyé avec un lien de réinitialisation. Après modification, le nouveau mot de passe est enregistré.
- **Affichage des données météorologiques** :
 - Le système demande des valeurs à la base de données (température, humidité, vent, précipitation).
 - Les valeurs sont utilisées pour afficher des graphiques.
- **Déconnexion et expiration de session** :
 - Une session peut expirer après 30 minutes d'inactivité (timer).
 - L'utilisateur peut aussi se déconnecter manuellement.

Conclusion :

Ce diagramme BPMN modélise un processus d'authentification et de gestion de compte utilisateur avec une interaction entre un serveur, un client web distant et un client web local. Il inclut également un module de récupération de mot de passe et d'affichage de données météorologiques.

Dictionnaire des acteurs

Nom de l'acteur	Description
Capteurs météorologiques	Envoient des données météorologiques.
Utilisateur sous Android	Visualise les données localement en temps réel et configure des seuils d'alerte.
Utilisateur Web local	Visualise l'historique des différentes grandeurs météorologique sous forme graphique ou sous forme de tableaux
Utilisateur Web Distant	Visualise l'historique des différentes grandeurs météorologique sous forme graphique ou sous forme de tableaux mais doit s'authentifier au préalable.

Description des cas d'utilisation

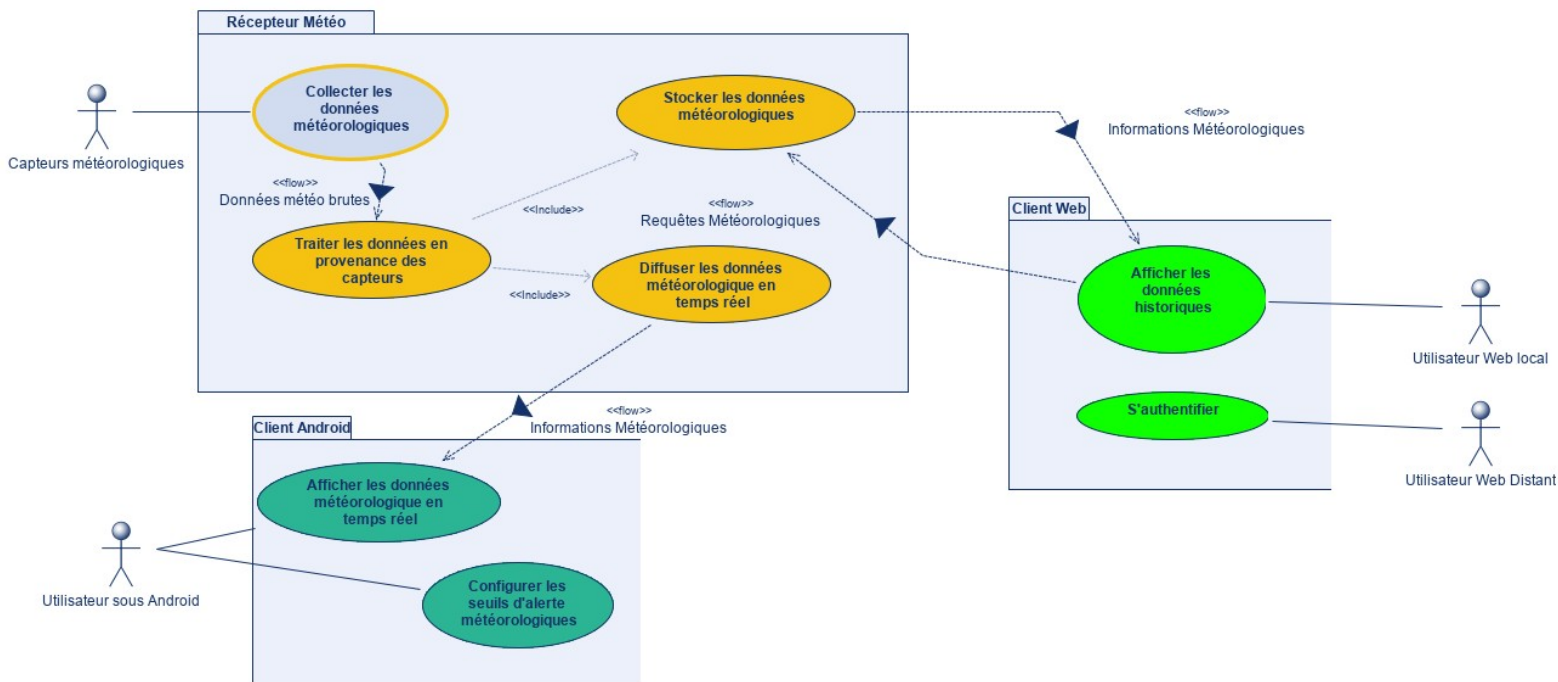


Image 4: Diagramme des cas d'utilisation

L'objectif de cette section est de présenter les cas d'utilisation en expliquant leurs finalités, leurs enjeux et leur contribution au bon déroulement du projet. Ils ont été créés à partir du diagramme des cas d'utilisation (ci-dessus) fourni au préalable dans le cahier des charges.

Chaque cas d'utilisation est décrit de manière claire et détaillée, en tenant compte des objectifs des utilisateurs, des données manipulées, ainsi que des contraintes ou des exceptions possibles. Cette approche vise à garantir une compréhension commune entre les différentes parties prenantes et à faciliter la conception, le développement et la validation du système.

Comme présenté ci-dessus chaque étudiant a étudié ses propres cas d'utilisation en fonction de ses objectifs au sein du projet.

Traiter les données en provenance des capteurs	
Description du cas d'utilisation	<p>Ce cas d'utilisation permet de recevoir les trames JSON issues de différents transmetteurs météorologiques via le programme RTL_433, de les analyser pour identifier les différents capteurs et en extraire les données météorologiques, puis d'exécuter deux actions principales :</p> <ol style="list-style-type: none"> 1. Alimenter une base de données en appelant le cas d'utilisation "Stocker les données météorologiques". 2. Transmettre les données aux clients Android en appelant le cas d'utilisation "Diffuser les données météorologiques en temps réel".
Pré-conditions	<p>Le programme RTL_433 est opérationnel et transmet les données JSON au récepteur météo.</p> <p>La configuration des capteurs (type, identifiant unique) est enregistrée dans le système, le format de la trame pour chaque transmetteur est connu.</p>
Scénario principal	<p>Réception des données:</p> <ul style="list-style-type: none"> ➤ Le système reçoit une trame JSON depuis RTL_433. ➤ La trame est analysée pour identifier le capteur à l'origine des données (via son identifiant ou un champ spécifique). Exemple de trame : {"time" : "2025-01-15 10:43:49", "model" : "Nexus-TH", "id" : 130, "channel" : 3, "battery_ok" : 1, "temperature_C" : 22.100, "humidity" : 31} <p>Extraction des données:</p> <ul style="list-style-type: none"> ➤ En fonction du type de transmetteur météo, les éléments de la trame (température, humidité, précipitations, vent, etc.) sont extraits et convertis si nécessaire (unités, formats). <p>Stockage des données</p> <ul style="list-style-type: none"> ➤ Pour chaque grandeur extraite (température, humidité, précipitations, vent), le système appelle le cas d'utilisation "Stocker les données météorologiques". <p>Diffusion des données</p> <ul style="list-style-type: none"> ➤ Les données extraites sont transformées en une trame JSON standardisée pour diffusion. ➤ Le système appelle le cas d'utilisation "Diffuser les données météorologiques en temps réel". Les données sont alors envoyées à tous les clients Android connectés.
Scénario alternatif	<p>Si une trame JSON est non reconnue (ne provient pas d'un de nos capteurs) ou invalide (possède des informations manquantes) elle est ignorée.</p>

Diagramme de séquence

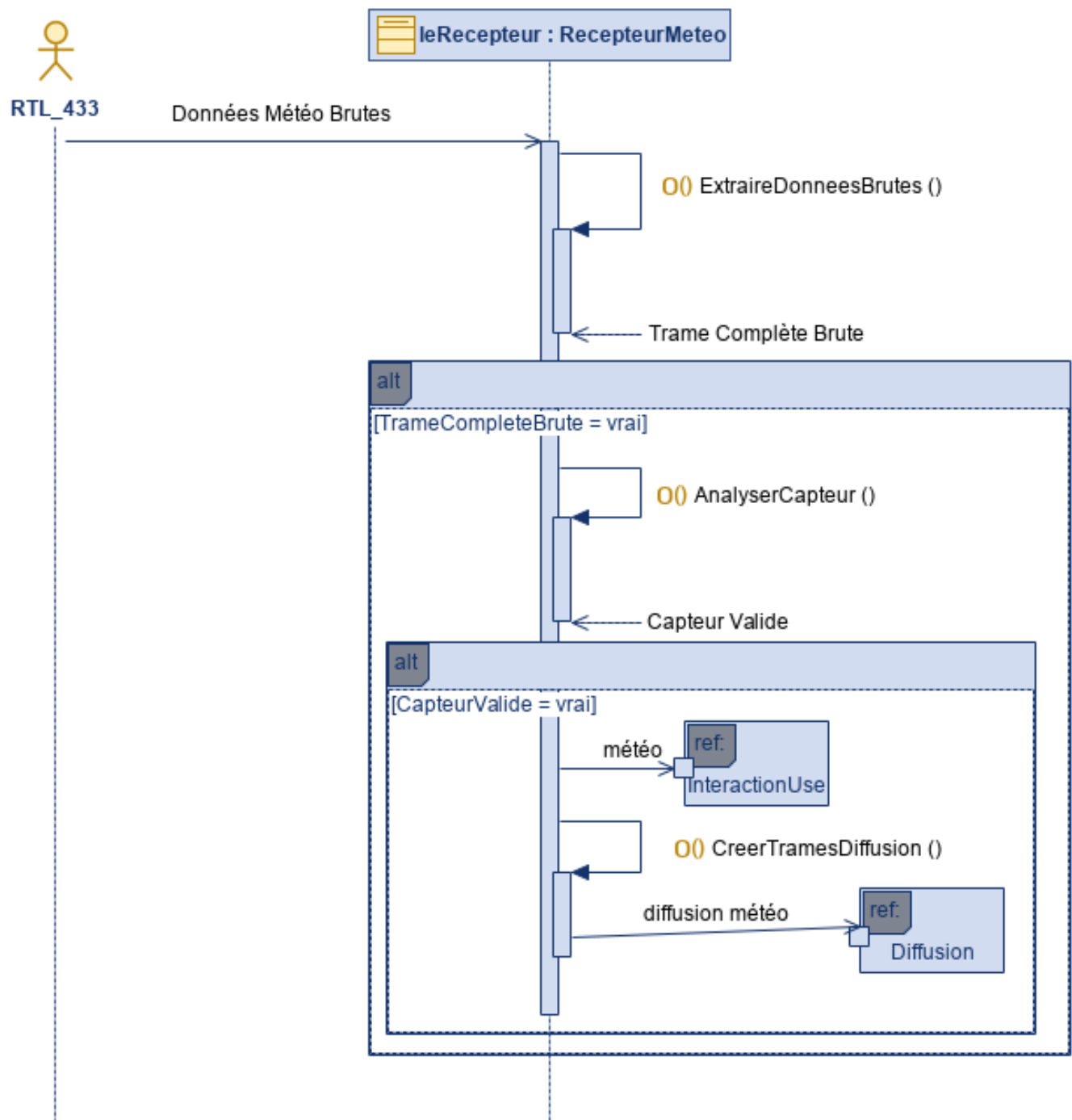


Image 5 : Diagramme de séquence "Traité les données"

Stocker les données météorologiques	
Description du cas d'utilisation	Ce cas d'utilisation permet de gérer l'enregistrement des données météorologiques extraites dans une base de données locale pour chaque grandeur mesurée (température, humidité, précipitations, vent). Les données stockées dans la base sont utilisées pour le cas d'utilisation « Afficher des données historiques ».
Pré-conditions	<p>La base de données locale est accessible et opérationnelle</p> <p>La structure des tables est correctement configurée: une table par type de grandeur, incluant les colonnes suivantes:</p> <ul style="list-style-type: none"> • Identifiant du capteur (entier) • Valeur mesurée (type selon la grandeur) • Horodatage
Post-conditions	<p>Les données météorologiques valides sont stockées dans la base de données.</p> <p>Les erreurs éventuelles sont enregistrées dans les journaux appropriés.</p>
Scénario Principal	<p>Réception des données</p> <ul style="list-style-type: none"> ➤ Les données extraites (grandeur, valeur, identifiant du capteur) sont reçues du cas d'utilisation "Traiter les données en provenance des capteurs". Le système identifie les tables qui vont faire l'objet d'une mise à jour. <p>Vérification de la dernière valeur enregistrée</p> <ul style="list-style-type: none"> ➤ Le système interroge chaque table concernée par le transmetteur météo afin de vérifier si la dernière valeur enregistrée est identique à celle reçue (même identifiant de du capteur et même valeur). <p>Enregistrement ou mise à jour des données</p> <ul style="list-style-type: none"> ➤ Si la nouvelle valeur diffère de la dernière enregistrée : Un nouvel enregistrement est ajouté à la table correspondante, incluant l'horodatage et l'identifiant du capteur et la nouvelle valeur. ➤ Si la nouvelle valeur est identique à la dernière enregistrée : L'horodatage de l'enregistrement existant est simplement mis à jour.
Scénarios Alternatifs	<p>Échec de la connexion à la base de données:</p> <ul style="list-style-type: none"> ➤ Si la base de données est inaccessible, le système retente la connexion après un délai défini. ➤ En cas d'échec persistant, les données sont temporairement stockées dans un fichier local ou une file d'attente. <p>Échec de l'insertion des données :</p> <ul style="list-style-type: none"> ➤ Si une erreur survient lors de l'insertion, le système l'enregistre dans un fichier local. <p>Incohérence dans les données reçues</p> <ul style="list-style-type: none"> ➤ Si les données reçues sont incomplètes, elles sont rejetées et enregistrées dans un journal d'erreurs pour analyse.

Diagramme de séquence

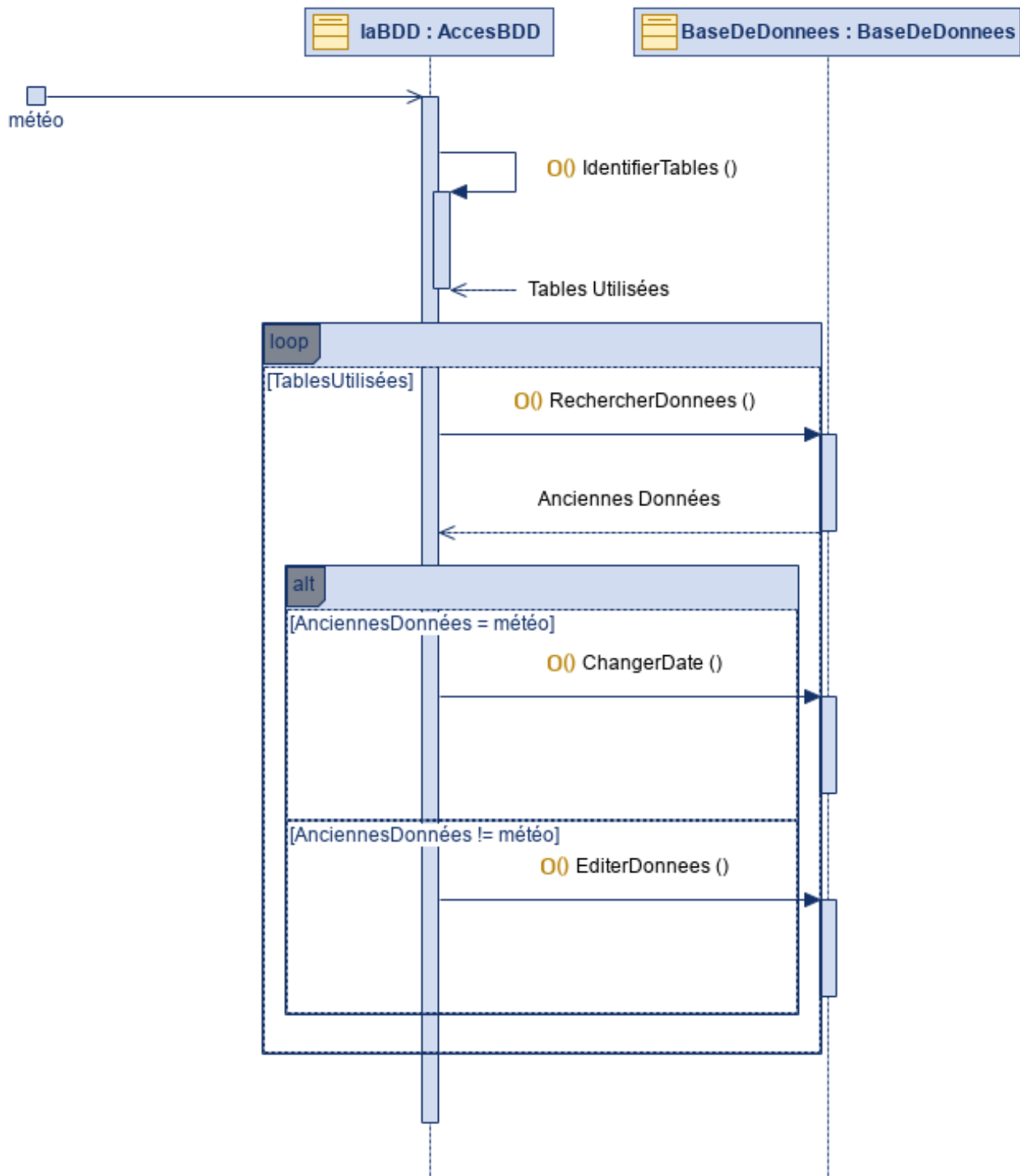


Image 6 : Diagramme de séquence “Stocker les données”

Diffuser les données météorologiques en temps réel	
Description du cas d'utilisation	Ce cas d'utilisation permet au système de réception météorologique d'envoyer, en temps réel, des trames standardisées au format JSON à tous les clients Android connectés.
Pré-conditions	Les données météorologiques doivent être standardisées dans un format JSON qui est connu des clients Android. Les clients abonnés doivent être connectés à la plateforme de diffusion, et disposés dans une liste active du système.
Scénario principal	<p>Préparation des données à diffuser :</p> <ul style="list-style-type: none"> ➤ Les données extraites (grandeurs, identifiant du capteur, horodatage) sont reçues du cas d'utilisation "Traiter les données en provenance des capteurs". ➤ Les données sont générées en JSON dans un format standardisé. Exemple de trame : <pre> { "capteur_id": "12345", "temperature": 25.3, "humidite": 65.2, "precipitations": 0.0, "vitesse_vent": 12.5, "horodatage": "2025-01-13T15:42:00" } </pre> <p>Diffusion des données :</p> <ul style="list-style-type: none"> ➤ Le système parcourt la liste des clients Android connectés. ➤ La trame JSON est envoyée à chaque client via une connexion réseau. <p>Le système enregistre un journal de diffusion.</p> <ul style="list-style-type: none"> ➤ La diffusion réussie (ou échouée) est inscrite dans un fichier journal ou un système de supervision. La réussite (ou l'échec) est reconnue grâce à une confirmation de réception en provenance des clients
Scénarios Alternatifs	<p>Client injoignable:</p> <ul style="list-style-type: none"> ➤ Si un client Android n'est pas joignable, trois tentative de réémission est effectuée. ➤ Si l'échec persiste, le client est marqué comme inactif et une alerte est enregistrée.

Diagramme de séquence

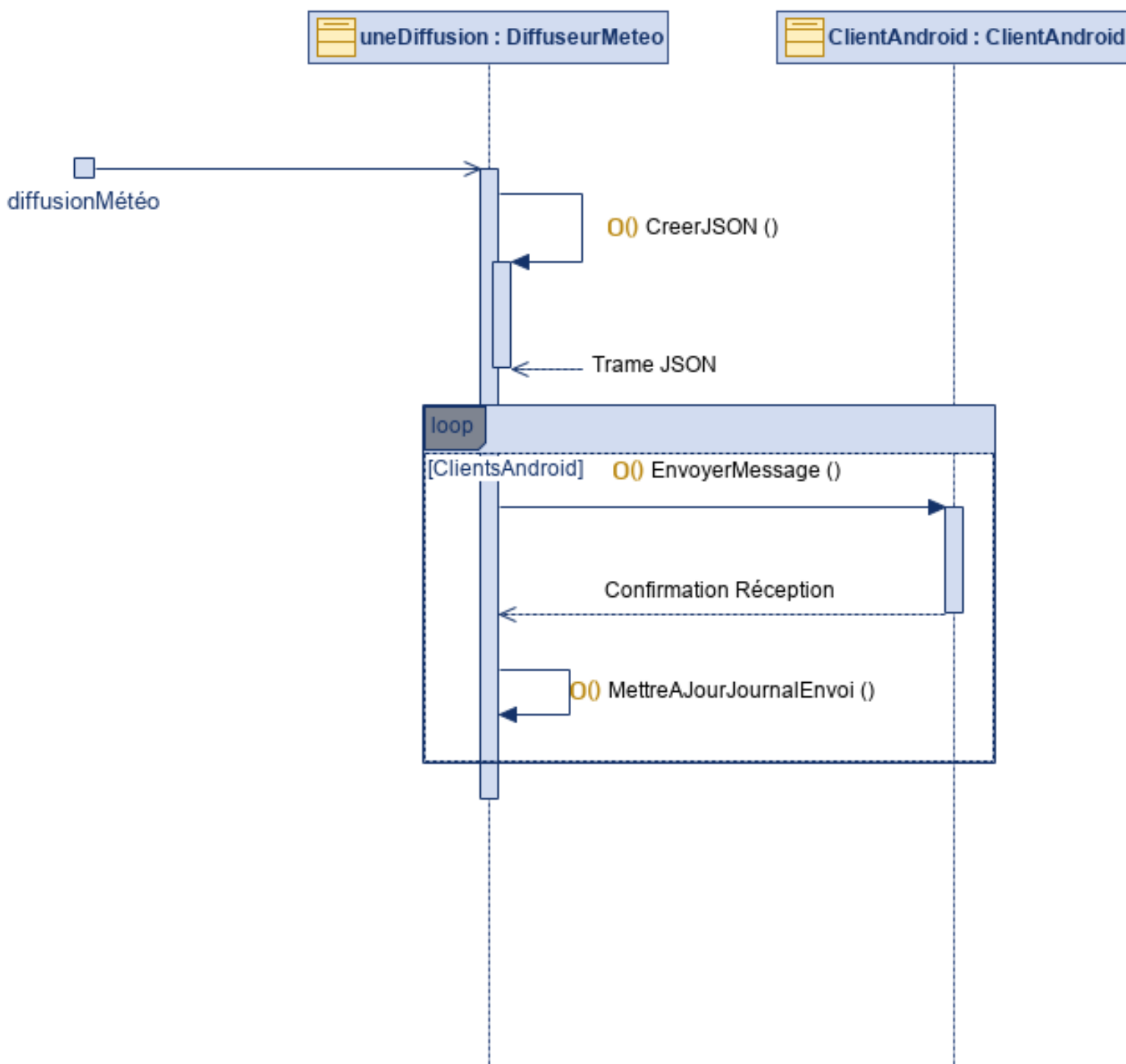


Image 7 : Diagramme de séquence "Diffuser les données"

Lors de notre analyse du diagramme de cas d'utilisation, nous avons pu nous rendre compte qu'il y avait d'autres paramètres à configurer donc nous avons modifiés "Configurer les seuils d'alerte météorologique" en "Configurer les paramètres météorologiques".

Configurer les paramètres météorologique	
Description	Ce cas d'utilisation permet à l'utilisateur de configurer des seuils d'alerte pour différentes grandeurs météorologiques (température, humidité, pluie, vent) directement sur l'application mobile. Lorsque les données reçues dépassent ces seuils, une notification est envoyée à l'utilisateur pour attirer son attention sur une condition météorologique importante. L'application mobile effectue l'analyse des données en temps réel pour vérifier si les seuils sont atteints.
Pré-conditions	L'utilisateur est connecté à l'application mobile et a accès à l'interface de configuration des alertes. L'application mobile reçoit en temps réel les trames JSON des données météorologiques du récepteur météo. L'utilisateur a défini au moins un seuil d'alerte pour l'une des grandeurs mesurées.
Post-conditions	Le système commence à surveiller les données météorologiques en fonction des seuils définis.
Scénario principal	<p>L'utilisateur Android ouvre l'application et accède à la section dédiée à la configuration des alertes météorologiques.</p> <p>L'utilisateur choisit les critères météorologiques à surveiller parmi les options disponibles, par exemple :</p> <ul style="list-style-type: none"> ➤ Température maximale ou minimale (ex. : > 35°C ou < 0°C) ➤ Vitesse du vent (ex. : > 50 km/h) ➤ Taux d'humidité (ex. : < 20% ou > 80%) ➤ Précipitations (ex. : 3mm) <p>Pour chaque critère sélectionné, l'utilisateur spécifie les valeurs seuils déclenchant une alerte, par exemple :</p> <ul style="list-style-type: none"> ➤ Température > 30°C : État alerte. ➤ Vitesse du vent > 60 km/h : État alerte. <p>L'utilisateur peut à tout moment revenir à l'interface pour ajuster les seuils ou réinitialiser les paramètres par défaut.</p> <p>L'utilisateur active ou désactive l'alerte pour chaque grandeur configurée.</p> <p>Pour chaque changement effectué, une notification de confirmation apparaîtra.</p> <p>Si une condition dépasse un seuil, une alerte est générée (notification)</p> <p>Dès qu'une trame JSON est reçue par l'application, celle-ci analyse les données en temps réel pour vérifier si l'une des grandeurs mesurées dépasse les seuils configurés. Si un seuil est atteint, l'application génère une notification de type "push".</p>

Diagramme de séquence

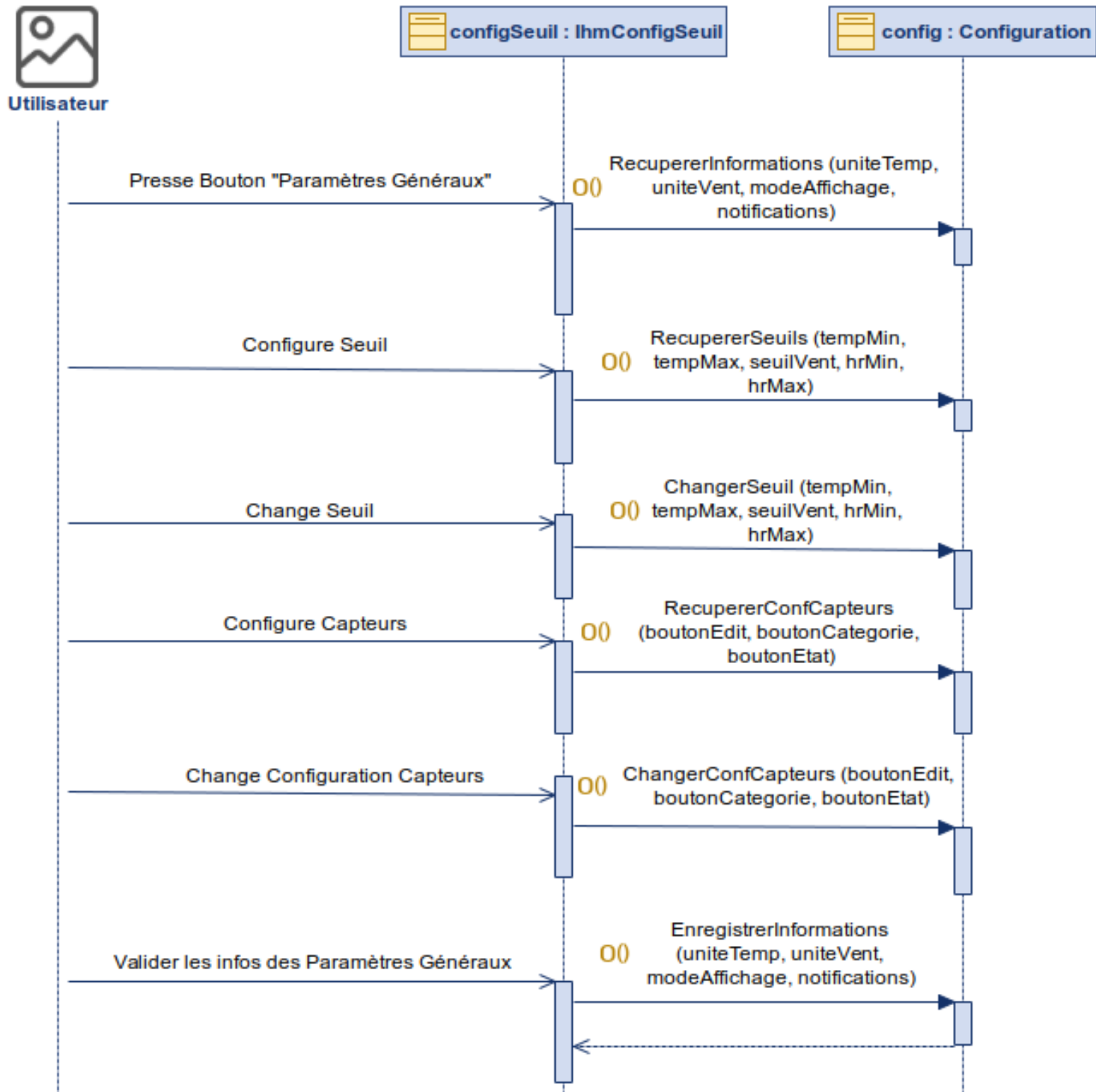


Image 8 : Diagramme de séquence "Configurer les paramètres Météorologiques"

Afficher les données météorologiques en temps réel	
Description	Ce cas d'utilisation permet d'afficher en temps réel les données météorologiques reçues par le récepteur météo et retransmises aux clients Android (tablettes ou téléphones). L'application offre une interface intuitive pour visualiser les informations essentielles avec des options de personnalisation.
Pré-conditions	Le client Android est connecté au réseau local. Les capteurs sont configurés avec leurs noms et types. La réception en temps réel des trames JSON est fonctionnelle.
Post-conditions	<ul style="list-style-type: none"> ➤ Les informations météo actuelles sont affichées à l'utilisateur (température, météo, humidité, vent, etc.)
Scénario principal	<p>Reçoit les informations du récepteur météo et les affiche à l'utilisateur sous Android.</p> <p>Elle récupère les informations principales, telles que :</p> <ul style="list-style-type: none"> ➤ Température actuelle : Chiffre décrivant la température (ex. : 26°). ➤ Conditions météo au-dessus de la température (ensoleillé, nuageux, pluvieux, etc.) ➤ Humidité représenté par un pourcentage ➤ Bouton pour réinitialiser la pluie et afficher la date de remise à zéro. ➤ Vitesse et direction du vent représenté par un graphique sous forme de rose des vents (de 0° à 360°). Où l'on voit des variations de force moyenne des vents selon leur orientation et par conséquent le repérage des vents dominants. ➤ Probabilité de précipitations représenté en pourcentage avec une courbe allant de 0 à 100. <p>Les données sont automatiquement actualisées à chaque réception d'informations.</p> <p>Ajout d'un mode graphique montrant l'évolution des grandeurs sur une période récente (ex. : dernière heure ou dernières 24 heures), avec des transitions fluides.</p> <p>Pouvoir paramétrer :</p> <ul style="list-style-type: none"> - Choix des unités de mesure (°C / °F, mm, km/h, m/s, mph...) - Notifications pour valeurs significatives (pluie ou vent fort) - Personnalisation des capteurs (renommage, catégories, activation / désactivation de l'affichage de certains capteurs) - Mode nuit pour une interface adaptée à la lecture dans l'obscurité avec des couleurs sombres
Alternative	<p>En cas de coupure de courant ou sans connexion internet, le système doit avertir l'utilisateur avec un message explicite, par exemple :</p> <p>"Impossible de récupérer les données météorologiques : Vérifiez votre connexion."</p>

Diagramme de séquence

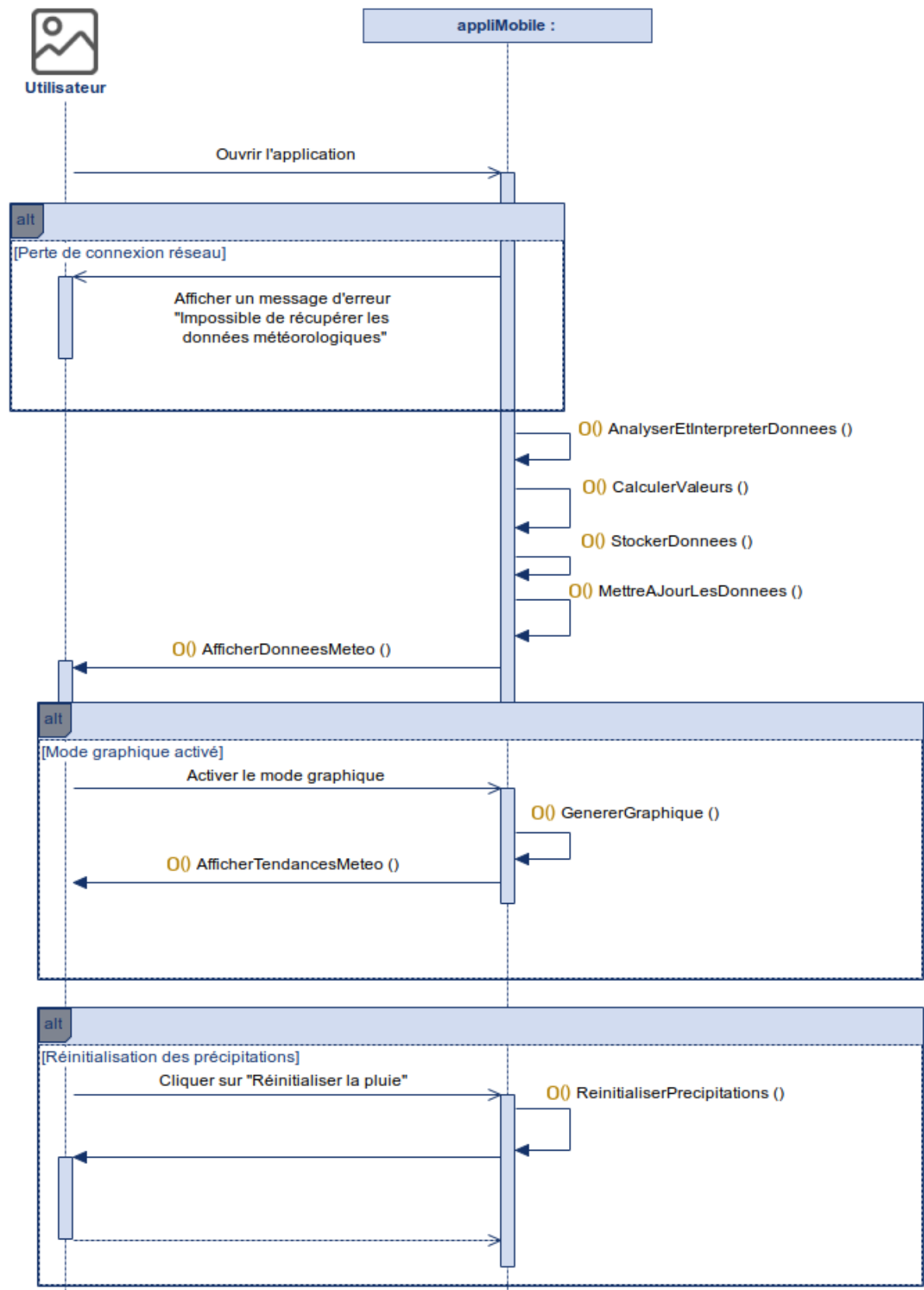


Image 9 : Diagramme de séquence “Afficher les données météorologiques en temps réel”

Afficher les données historiques	
Description du cas d'utilisation	<p>Ce cas d'utilisation permet aux utilisateurs du site web de visualiser les données météorologiques historiques collectées par le système. Les données sont affichées sous différentes formes selon leur type (courbes, barres graphiques, diagrammes polaires) pour faciliter leur interprétation. Une interface utilisateur intuitive permet de :</p> <ul style="list-style-type: none"> > Sélectionner le type de grandeur météorologique (température, humidité, précipitations, vent). > Zoomer sur des périodes spécifiques (dernier mois par défaut, avec possibilité d'extension ou de restriction). > Exporter les données dans un format compatible avec les tableurs pour une analyse avancée.
Pré-conditions	Être, au préalable, connecter à la page web
Scénario principal	<p>L'utilisateur décide en appuyant sur des boutons ce qu'il veut afficher et en fonction de ce qui est cliqué :</p> <ol style="list-style-type: none"> 1. Graphique polaire du vent : Une représentation graphique sous forme de cercle (avec les 16 positions différentes de la girouette). Où l'on voit des variations de force moyenne des vents selon leur orientation et par conséquent le repérage des vents dominants. Leur orientation sera définie dans la base de données et sera retranscrit dans le graphique sous forme de colonne et la force du vent sera représenté sur le même graphique mais sous forme de zone 2. Température : courbe (line charts) représentant les températures enregistrées et en fonctions de ces températures la couleur de la courbe change : <ul style="list-style-type: none"> > bleu pour température froid > vert pour température "idéale" > rouge pour température chaude 3. Humidité : graphique évoluant en fonction de l'humidité avec différentes couleurs selon l'humidité présente allant du bleu clair au violet. En fonction du capteur et de la base de données, savoir repérer l'id pour informer l'utilisateur en fonction de l'endroit du capteur (extérieur, intérieur, serre) 4. Précipitation: diagramme en barre (plaine) représentant les précipitations avec différentes couleurs. La couleur change en dégradé du bleu ciel au violet en fonction du niveau de précipitation. <p>Sur tous les graphiques l'utilisateur peut définir la période sur laquelle il veut avoir ses informations (années, mois, semaines, jours). Sur tous les graphiques les données de tous les capteurs seront inscrites et enregistrées dans la base données. Ces données devront être extraites de la base et adaptées de sorte à avoir des points par exemple dans la base de données l'humidité présente dans l'air est de 40% de 16h à 16h16 mais n'est présent que la données à 16h16. Le graphique doit le prendre en compte et adapter le graphique en recréant les points toute les 30 sec de 16h00 à 16h16</p>

Diagramme de séquence

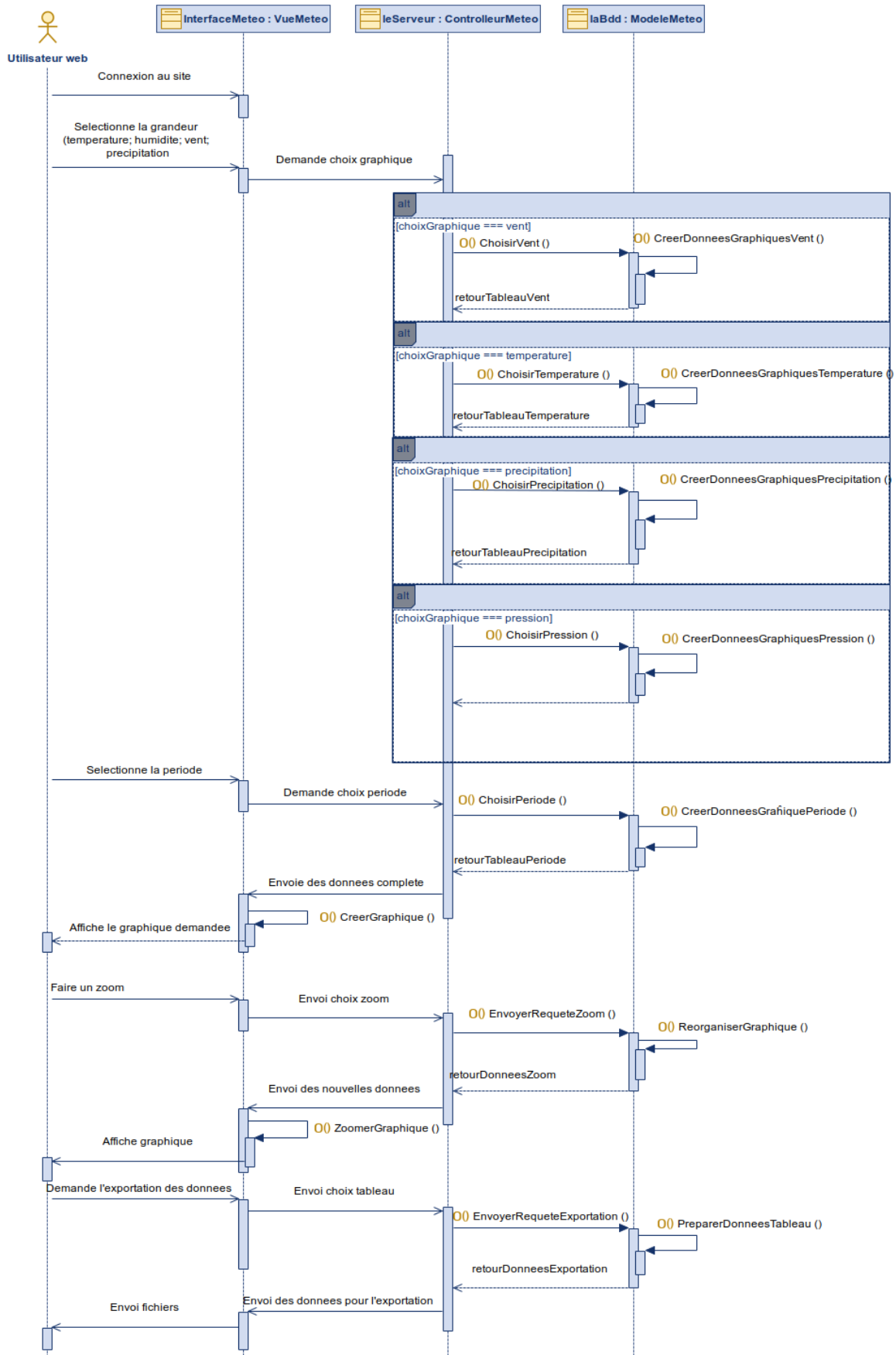


Image 10 : Diagramme de séquence "Afficher données historiques"

S'authentifier	
Description du cas d'utilisation	Ce cas d'utilisation permet à un internaute situé en dehors du réseau local d'accéder aux fonctionnalités du site web (notamment "Afficher les données historiques"). L'utilisateur doit fournir des identifiants valides pour s'authentifier. Une fois connecté, il peut naviguer dans les pages du site et accéder aux données autorisées.
Pré-conditions	<ul style="list-style-type: none"> ➤ L'utilisateur doit disposer d'un compte enregistré avec un identifiant (adresse e-mail ou nom d'utilisateur) et un mot de passe. ➤ Le système doit être configuré pour différencier les connexions locales des connexions externes.
Post-conditions	<p>Si l'authentification est réussie :</p> <ul style="list-style-type: none"> ➤ L'utilisateur est authentifié et a accès à son espace personnel ou aux fonctionnalités autorisées. ➤ Une session est créée pour l'utilisateur. <p>Si l'authentification échoue :</p> <ul style="list-style-type: none"> ➤ L'utilisateur reste non authentifié. ➤ Un message d'erreur approprié est affiché. ➤ Si l'utilisateur est nouveau lui faire créer une authentification puis l'enregistrer. ➤ Une session sécurisée est active pour l'utilisateur authentifié. ➤ Le système enregistre la date et l'heure de la connexion pour des raisons de traçabilité.
Scénario principal	<p>Accès à la page de connexion</p> <ul style="list-style-type: none"> ➤ L'internaute accède à l'URL du site web depuis un réseau externe. ➤ Il est automatiquement redirigé vers une page de connexion si non authentifié. <p>1. Saisie des informations d'identification</p> <ul style="list-style-type: none"> ➤ L'utilisateur entre son identifiant et son mot de passe dans les champs prévus. <p>2. Validation des informations</p> <p>Le système vérifie que :</p> <ul style="list-style-type: none"> ➤ L'identifiant existe dans la base de données des utilisateurs. ➤ Le mot de passe correspond à celui enregistré pour cet identifiant. <p>3. Accès autorisé</p> <p>Si les informations sont correctes :</p> <ul style="list-style-type: none"> ➤ L'utilisateur est redirigé vers la page d'accueil ou la page qu'il tentait d'accéder. ➤ Une session sécurisée est créée via un cookie ou un token. <p>4. Accès refusé</p> <p>Si les informations sont incorrectes :</p> <ul style="list-style-type: none"> ➤ Un message informe l'utilisateur que les identifiants sont

	<p>invalides.</p> <ul style="list-style-type: none">➤ Une option est proposée pour réinitialiser le mot de passe en cas d'oubli.
Alternatives	<p>Mot de passe oublié</p> <ul style="list-style-type: none">➤ L'utilisateur clique sur un lien "Mot de passe oublié".➤ Le système lui demande son adresse e-mail.➤ Un lien de réinitialisation est envoyé par e-mail.➤ L'utilisateur peut définir un nouveau mot de passe après avoir cliqué sur ce lien. <p>Tentatives multiples échouées</p> <ul style="list-style-type: none">➤ Après plusieurs tentatives échouées, le compte de l'utilisateur est temporairement verrouillé pour éviter les attaques par force brute.➤ Un message indique la durée du verrouillage ou la procédure pour débloquer le compte. <p>Session expirée</p> <p>Si la session utilisateur expire après une période d'inactivité, l'utilisateur doit se reconnecter pour continuer à naviguer.</p>

Diagramme de séquence

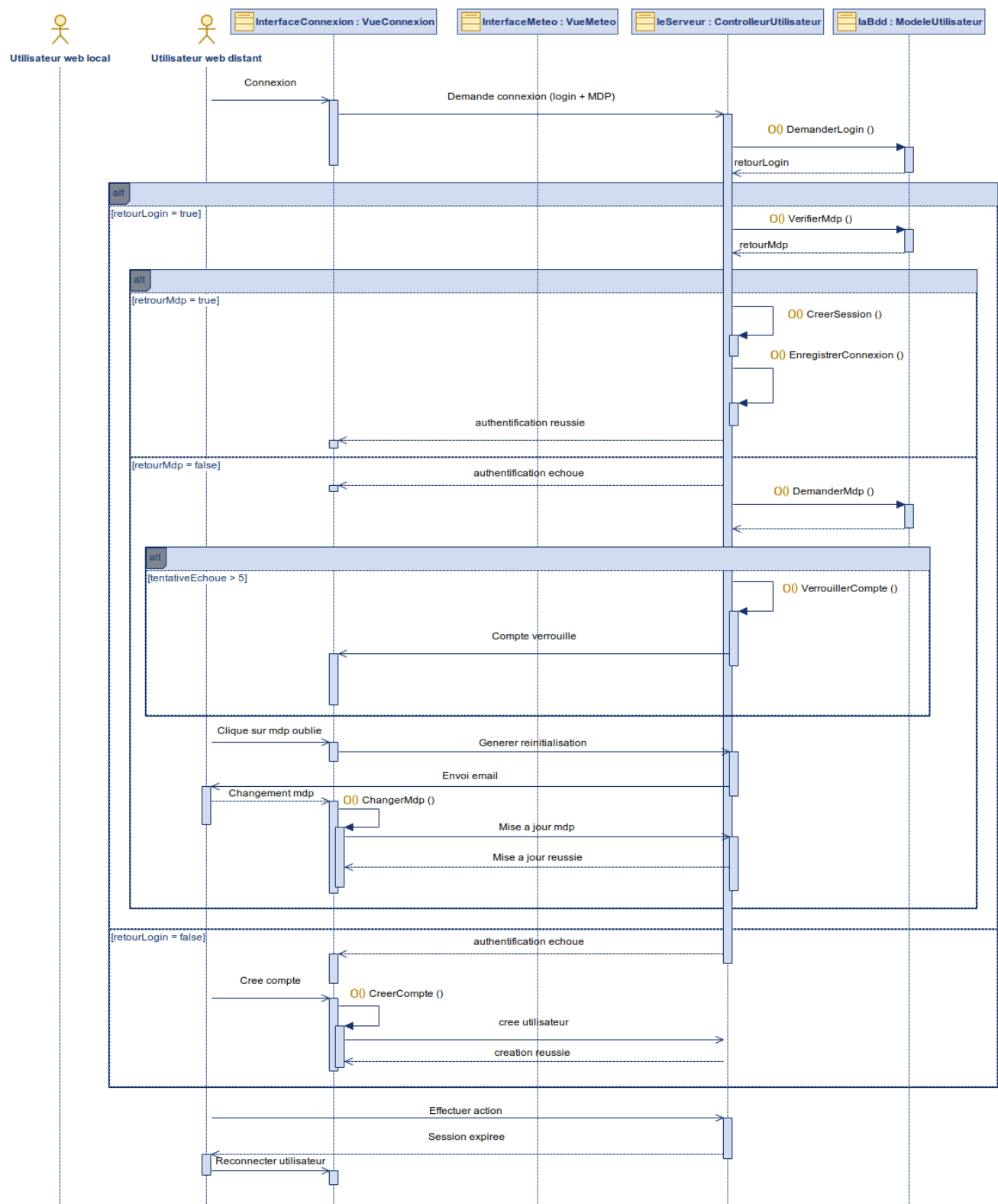


Image 11 : Diagramme de séquence “s’authentifier”

Matrice RACI

Processus	Récepteur Météo	Base de données	Client Android	Client Web
Traiter les données	R	-	-	-
Stocker les données	R	A	-	-
Diffuser les données	C	-	A	-
Afficher les données en temps réel	-	-	R	-
Configurer les seuils d'alerte	-	-	R	-
Afficher les données historiques	-	C	-	R
S'authentifier	-	A	-	R

La matrice RACI permet d'identifier la responsabilité de chacun des acteurs au sein de chacun des cas d'utilisation présentés précédemment.

Chaque lettre représente un niveau de responsabilité:

- **R** : Responsable (effectue l'action).
- **A** : Autorité (prend les décisions finales).
- **C** : Consulté (donne un avis ou fournit des informations).
- **I** : Informé (est tenu au courant des résultats)

Diagrammes de classe
Le diagramme de classe permet d'introduire les premières fonctions et les premiers attributs des classes. Ce diagramme n'est pas définitif, celui-ci viendra à évoluer en fonction des outils que nous allons choisir lors de la phase de conception. La description de ces diagrammes sera donc généralisée.

Récepteur Météorologique

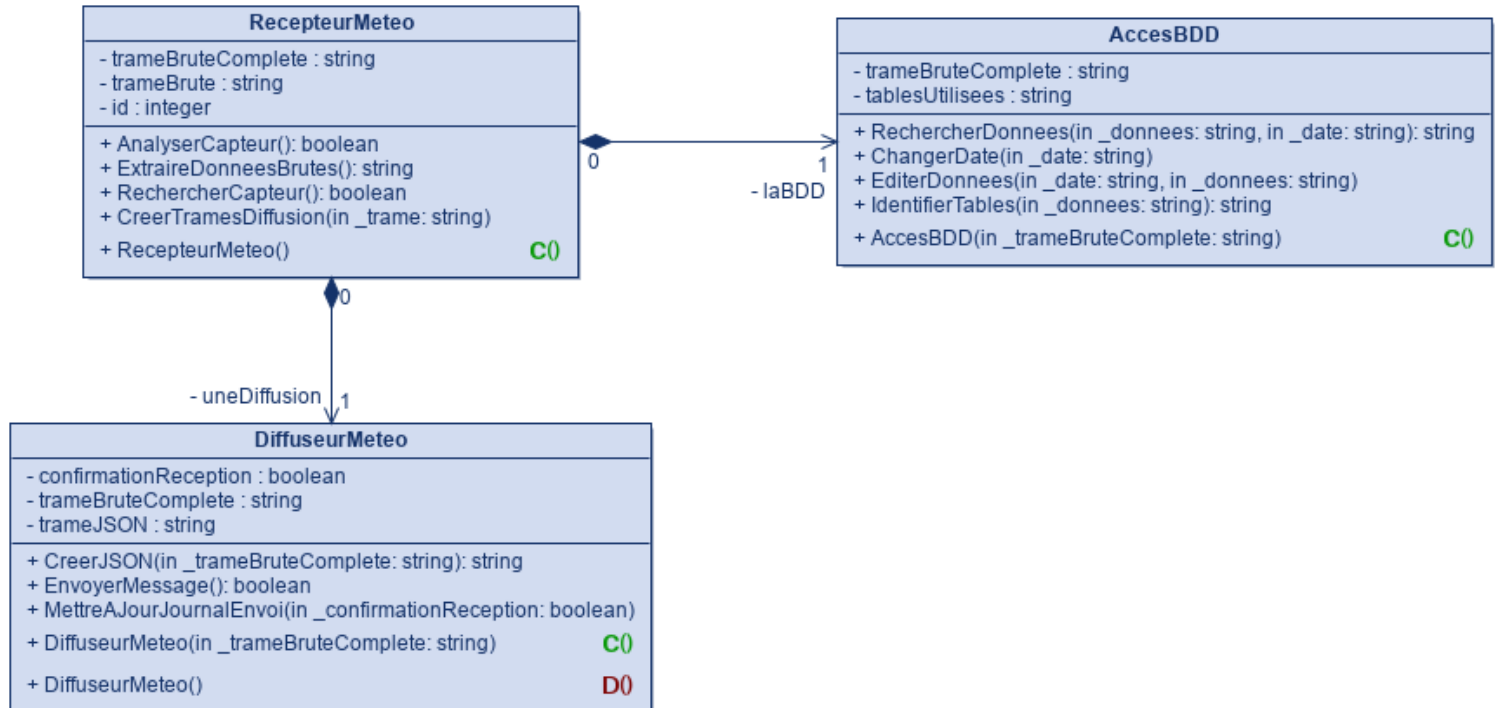


Image 12: Diagramme de classe "Récepteur Météorologique"

Classe **RecepteurMeteo**:

Cette classe va nous permettre de lire les trames reçues des capteurs. C'est la classe principale du programme du récepteur, elle a donc une relation de composition avec les 2 autres classes. Cette première classe identifiera le capteur et préparera les données pour les autres classes.

Classe **DiffuseurMeteo**:

Cette classe va nous permettre de diffuser les informations aux clients Android. Les données seront traitées par la classe **RecepteurMeteo**, cette classe n'aura plus qu'à créer les trames JSON et à les diffuser. Elle s'occupera aussi de la mise à jour du journal d'activité.

Classe **AccesBDD**:

Cette classe va nous permettre de stocker les données dans la base de données. Les données seront traitées par la classe **RecepteurMeteo**, cette classe n'aura plus qu'à envoyer une requête SQL afin d'enregistrer les données dans la base.

Contrôleur Météorologique

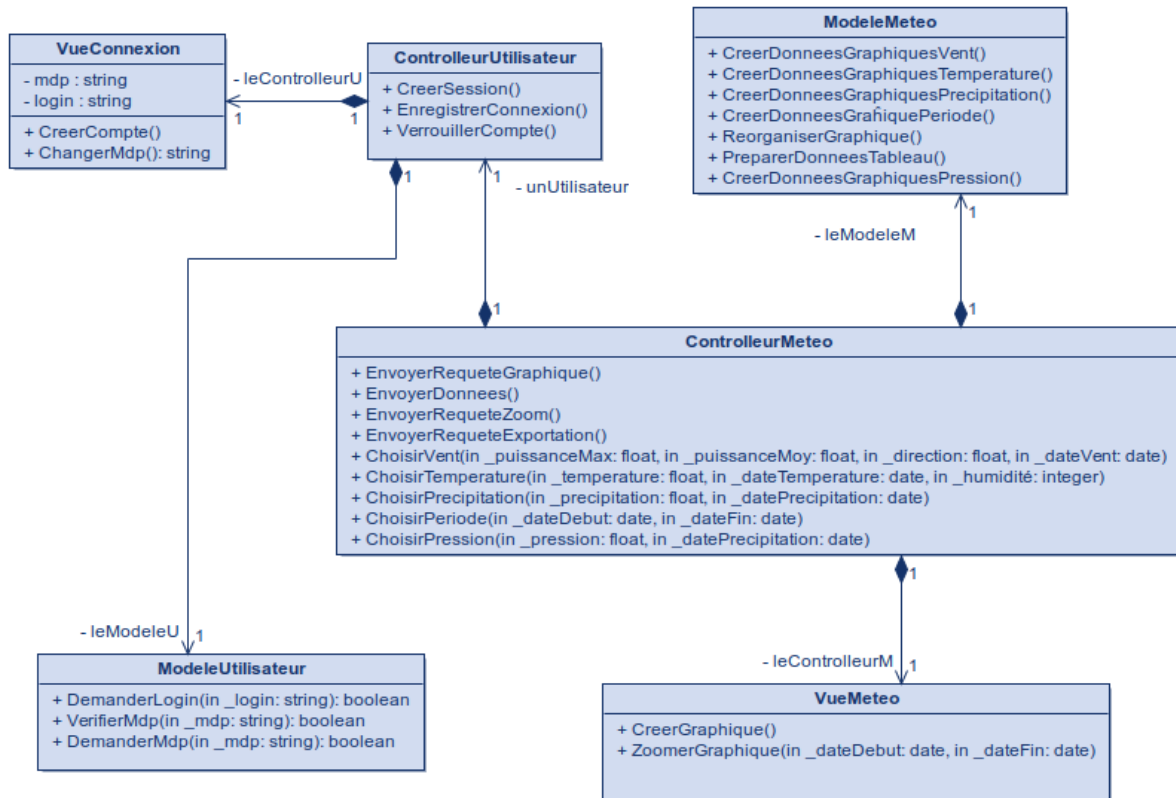


Image 13: Diagramme de classe "Client Web"

1. VueConnexion

Cette classe représente l'interface utilisateur pour la connexion et la gestion du compte. Elle permet aux utilisateurs de s'authentifier, de créer un compte et de changer leur mot de passe. Son rôle est d'afficher les écrans de connexion et d'envoyer les informations saisies à l'application pour vérification.

2. ControleurUtilisateur

Le contrôleur utilisateur gère la logique de connexion et de gestion des comptes. Il agit comme un intermédiaire entre la **VueConnexion** et le **ModeleUtilisateur**. Son rôle est d'authentifier les utilisateurs, d'ouvrir ou de verrouiller une session, et d'enregistrer les connexions.

3. ModeleUtilisateur

Cette classe est responsable de la gestion des données utilisateurs. Elle contient les informations d'authentification et permet de vérifier les identifiants et mots de passe. C'est elle qui valide si un utilisateur peut se connecter ou non. Elle stocke aussi les comptes utilisateurs.

4. ControleurMeteo

Ce contrôleur gère toutes les interactions liées aux données météorologiques. Il agit comme un intermédiaire entre la **VueMeteo** et le **ModeleMeteo**. Son rôle est de traiter les requêtes pour afficher des graphiques, envoyer des données, appliquer des filtres et exporter les informations météorologiques. Il s'assure que les bonnes données sont demandées et affichées correctement.

5. ModeleMeteo

C'est la partie qui gère les données météorologiques. Elle récupère, traite et organise les informations comme la température, le vent, la pression et les précipitations. Elle permet aussi de créer des représentations graphiques de ces données. Son rôle est donc de préparer les informations avant de les envoyer à la **VueMeteo**.

6. VueMeteo

Cette classe représente l'interface utilisateur pour l'affichage des données météorologiques. Elle permet de visualiser les graphiques et de zoomer sur des périodes spécifiques. Elle récupère les informations depuis le **ControleurMeteo** et les affiche de manière compréhensible pour l'utilisateur.

Client Android

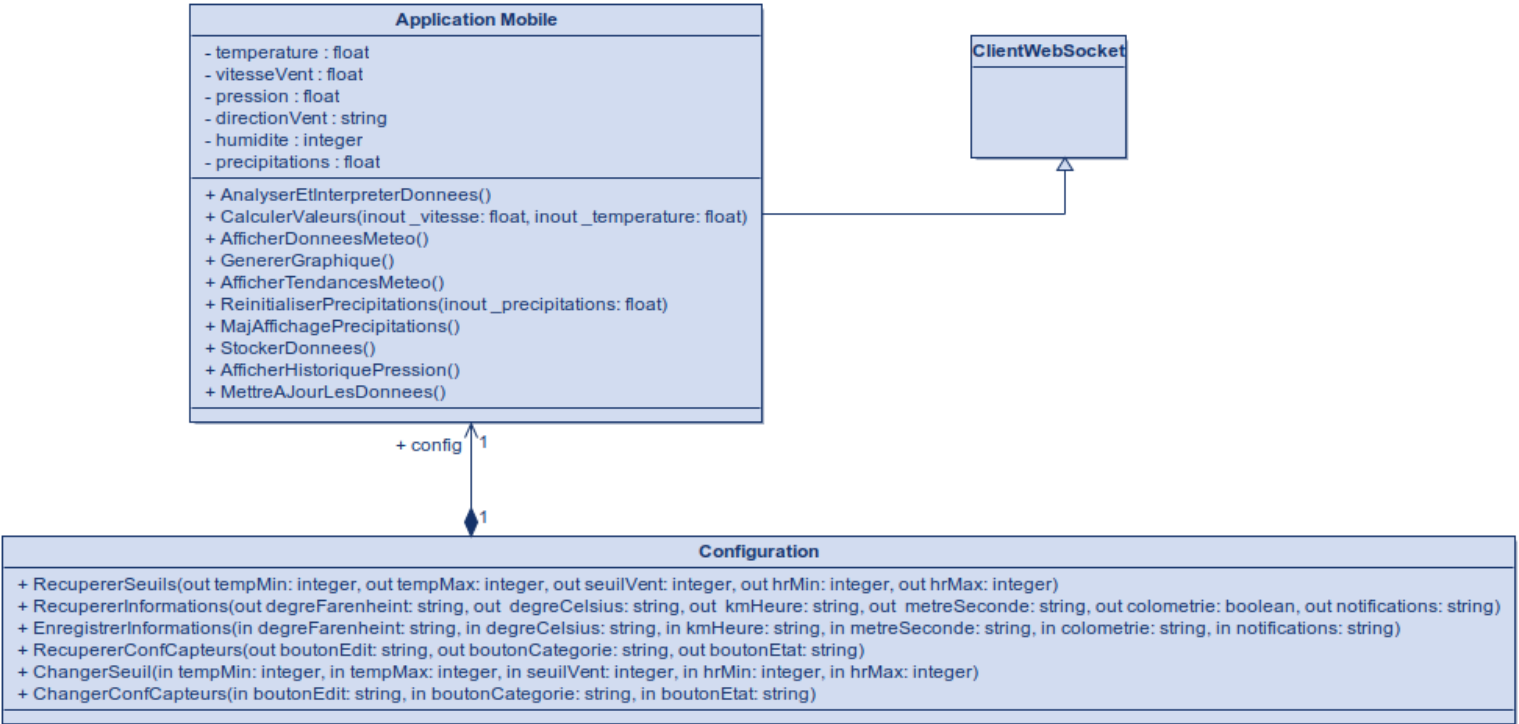


Image 14: Diagramme de classe “Client Android”

1. Application Mobile

Cette classe représente l’interface principale du Client Android lui permettant d’accéder à toutes les données nécessaires.Cette classe reçoit, traite et affiche les données météorologiques que le serveur envoie.

2. Configuration

Cette classe gère la configuration des seuils et des capteurs de l’application, enregistre les préférences (°C, °F, m/s) et permet d’adapter les notifications et alertes en fonction des préférences de l’utilisateur

3. ClientWebSocket

Cette classe permettra au Client Android de pouvoir se connecter au serveur récupérant les différentes données nécessaires à son utilisation.

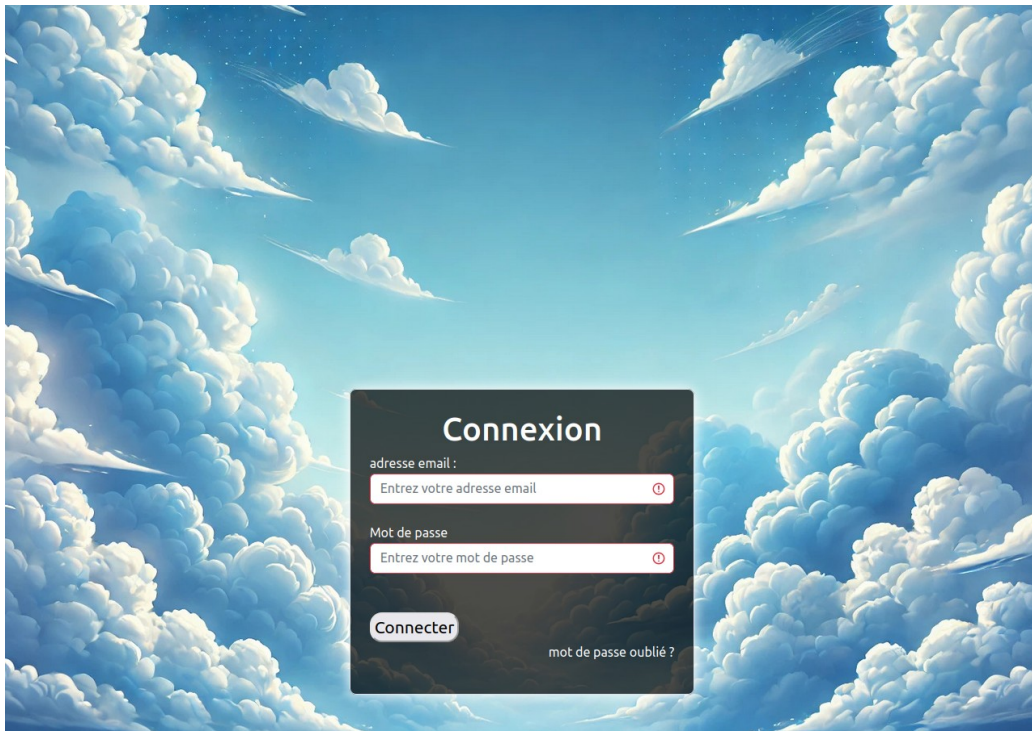
IHM**Client Web**

Image 15 : Page Web “Page de Connexion”

La page de connexion sert à chaque utilisateur distant de se connecter au serveur web. L'utilisateur doit au préalable se connecter en entrant son nom d'utilisateur (adresse email) et en entrant son mot de passe.

Un bouton sera disponible si l'utilisateur a oublié son mot de passe et un autre sera disponible si l'utilisateur n'a pas encore de compte.



Image 16 : Page Web "Afficher les Données"

Après s'être connecter les utilisateurs seront accueilli sur la page météo où les graphiques s'afficheront dont le premier qui seras le dernier regarder en date. On pourra le savoir grâce aux cookies. Chaque graphique est disponible sur la barre de recherche mais un seul peut-être affiché. Un fond uni sera mit afin de mettre un peu de vie à la page.

Client Android

Menu principal

Interface principale du Client Android permettant de savoir la température actuelle, et avoir un suivi des différentes informations météorologiques tel que la vitesse (en km/h ou m/s) et la direction du vent, la température intérieur/extérieur (en °C ou en °F), les précipitations en mm puis la pression en hpa (hectopascal). Grâce à ce menu, il est possible d'accéder directement aux différents menus disponibles.

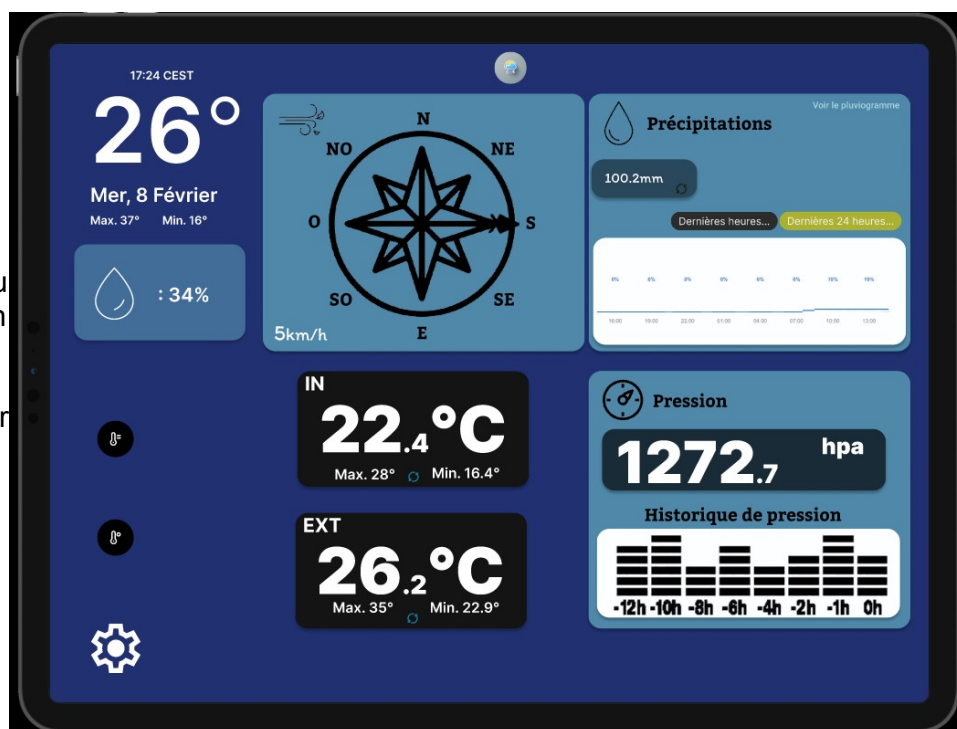


Image 17 : Menu principal application Android

Menu température

Ce menu permet de voir la courbe de température en fonction de la journée choisie, par exemple, le Lundi 1 juin 2033, il y aura une température maximum de 26°C et de minimum 14°C et une température actuelle de 18°C.



Image 18 : Menu température application Android

Menu capteurs

Ce menu permet de voir les différentes informations reçues des différents capteurs de température connectés à votre serveur.

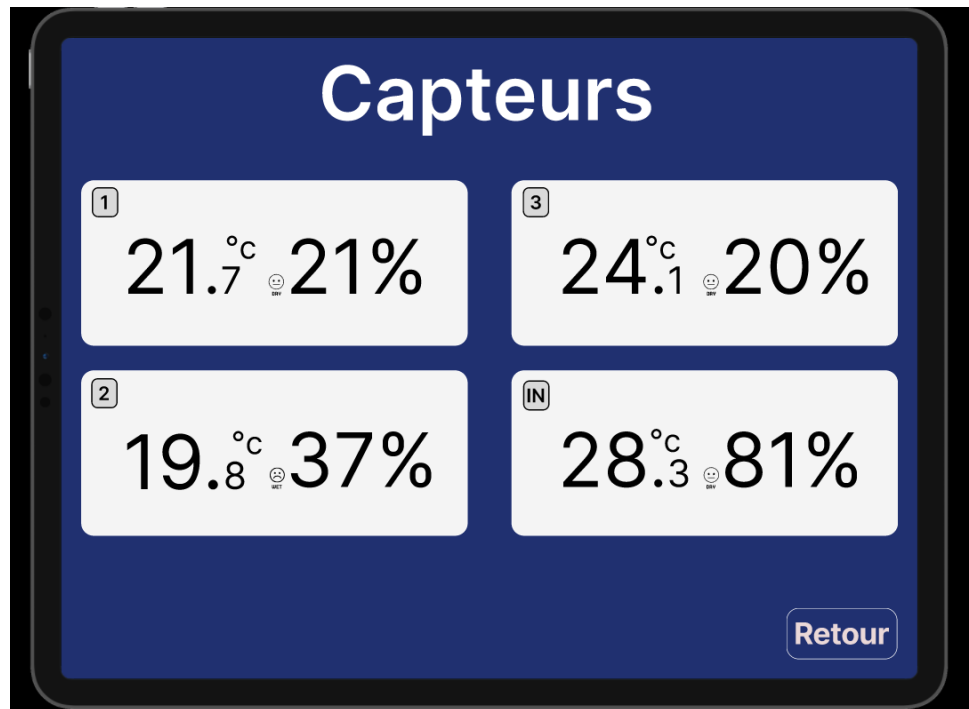


Image 19 : Menu capteurs application Android

Menu paramètres généraux

Ce menu permet de changer les unités (par exemple km/h en m/s), de changer l'interface en mode sombre ou éclairée, d'accepter la réception de notification et d'accéder au menu des réglages des capteurs et le menu des configurations des seuils. Ces paramètres seront actifs une fois que le bouton "Valider" est pressé.



Image 20 : Menu paramètres généraux application Android



Menu réglages des capteurs

Ce menu permet de changer les différents paramètres des capteurs tel que :

- son nom
- sa catégorie

Il est aussi possible de désactiver les capteurs

Image 21 : Menu réglages des capteurs application Android

Menu configuration des seuils

Ce menu permet de choisir les différents seuils. Ils sont applicables à la température, au vent et au taux d'humidité. Ces paramètres seront actifs une fois que le bouton "Valider" est pressé.



Image 22 : Menu configuration des seuils généraux application Android

Pluviogramme

Cette interface permet d'afficher une courbe représentant les précipitations sur une durée définie au préalable.

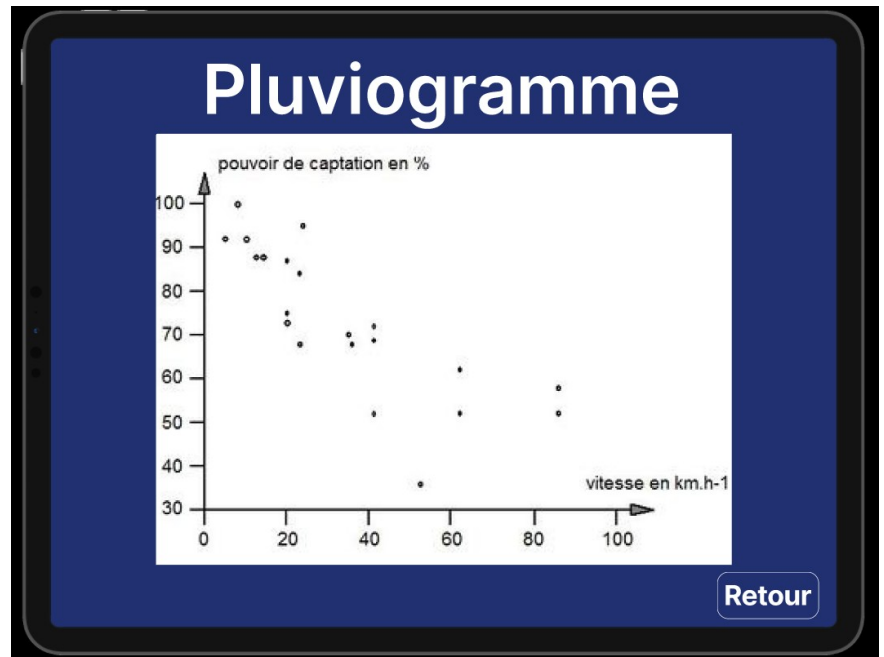


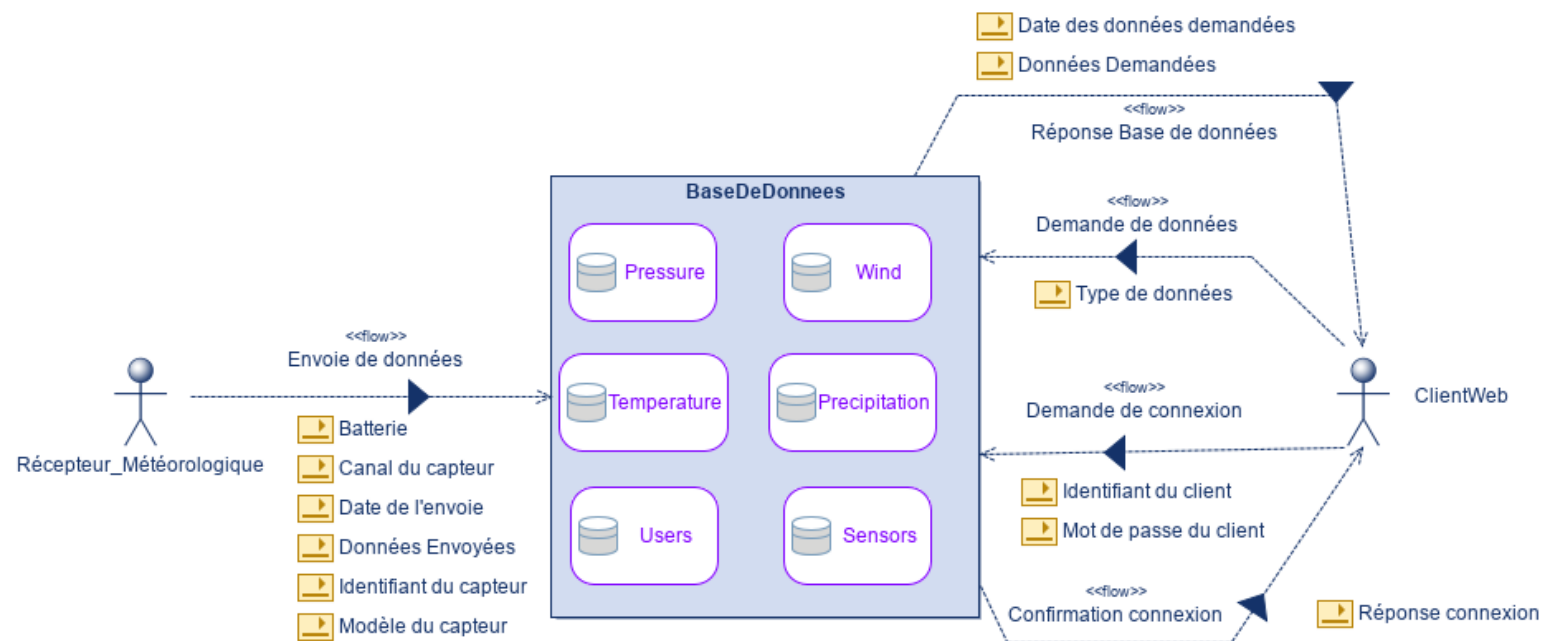
Image 23 : Interface pluviogramme application Android

Diagramme de flux de données

Ce diagramme de flux de données (DFD) illustre le fonctionnement du système de gestion des données météorologiques, conçu pour enregistrer, stocker et fournir des informations climatiques précises. Le système central repose sur une base de données qui sauvegarde des paramètres tels que la température, l'humidité, la direction et la vitesse du vent, ainsi que les précipitations.

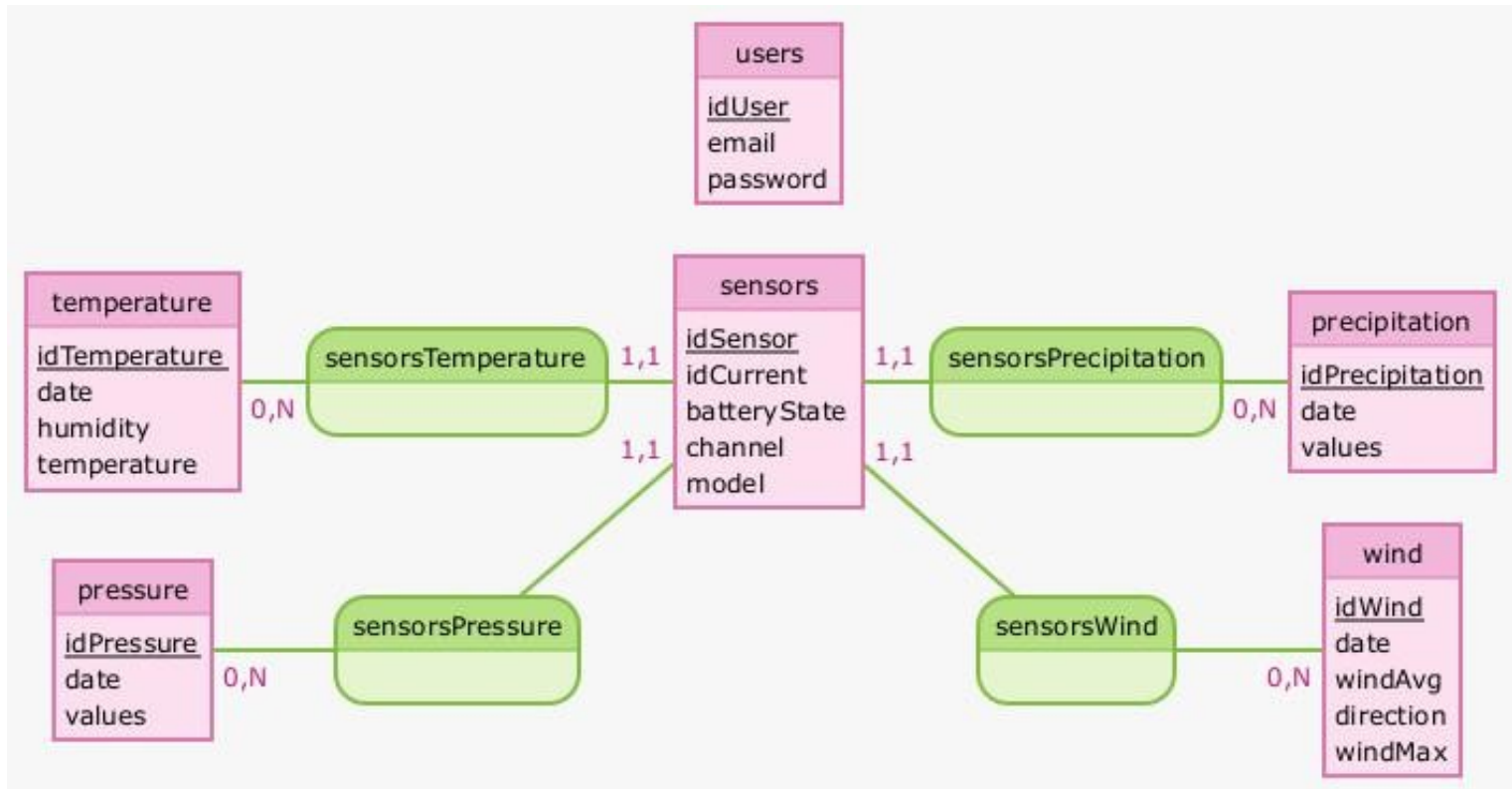
Le DFD montre :

- Les flux de données provenant du récepteur, incluant les différents paramètres climatiques envoyés au système.
- Les tables internes à la base de données, qui permettent la gestion et le stockage des données.
- Les flux de données sortants vers l'utilisateur, comprenant les résultats des requêtes formulées (historiques des données météo).



- Les interactions d'identification effectuées par l'utilisateur pour accéder au système.

Image 24 : "Diagramme de flux de données"



Description de la base de données

Image 25 : "Diagramme MCD"

Cette image représente la base de données correspondante à notre projet. Celle-ci est composée de plusieurs tables dont voici la description:

1. Users

- **idUser** : Désigne l'identifiant unique pour chaque utilisateur (clé primaire).
- **email** : Désigne l'adresse électronique de l'utilisateur.
- **password** : Désigne le mot de passe de l'utilisateur pour la connexion.

2. Temperature

- **idTemperature** : Désigne l'identifiant unique de chaque capteur (clé primaire).
- **date** : Désigne la date associée à l'enregistrement de la température et de l'humidité.
- **humidity** : Désigne les valeurs de l'humidité mesurée.
- **temperature** : Désigne les valeurs des températures mesurées.

3. Precipitation

- **idPrecipitation** : Désigne l'identifiant unique de chaque capteur (clé primaire).
- **date** : Désigne la date associée à l'enregistrement de la précipitation.
- **values** : Désigne les valeurs de précipitations mesurées.

4. Wind

- **idWind** : Désigne l'identifiant unique de chaque capteur (clé primaire).

- **date** : Désigne la date associée à l'enregistrement du vent.
- **wind_Avg** : Désigne la vitesse moyenne du vent.
- **direction** : Désigne la direction du vent.
- **wind_Max** : Désigne la vitesse maximale du vent durant l'enregistrement du capteur.

5. Pressure

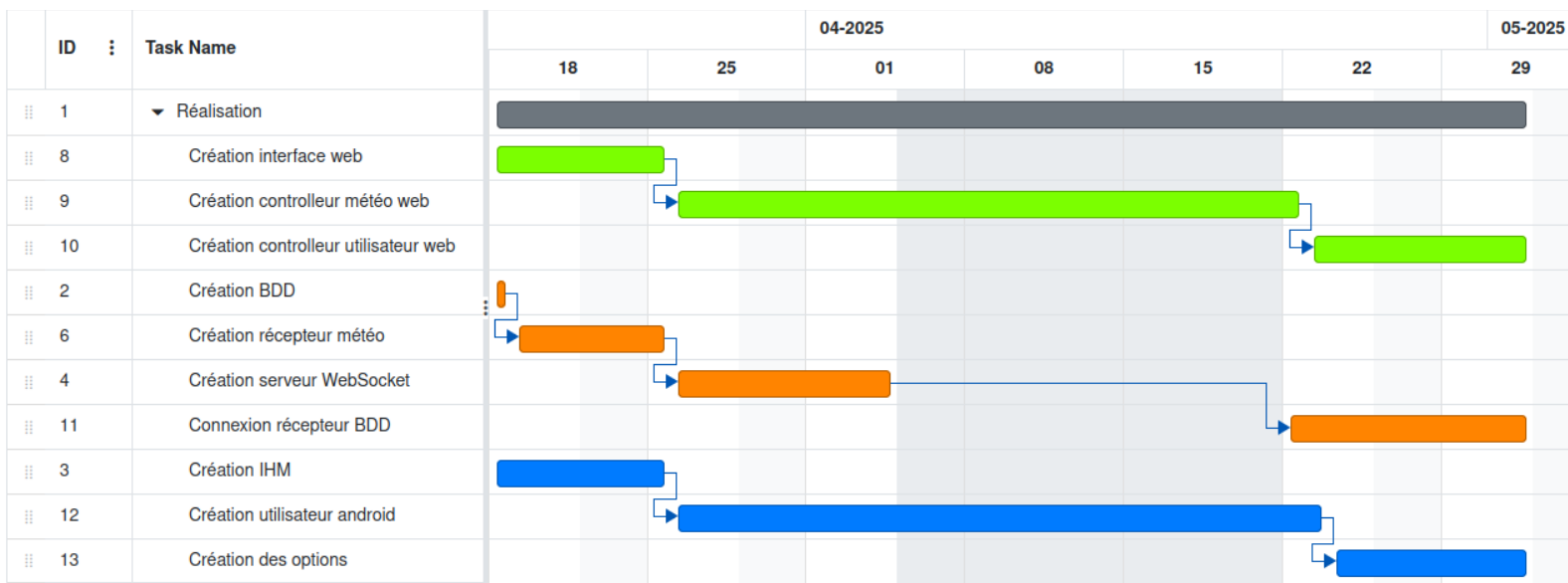
- **idPressure** : Désigne l'identifiant unique de chaque capteur(clé primaire).
- **date** : Désigne la date associée à l'enregistrement de la pression de l'air.
- **values** : Désigne la valeur de la pression mesurée.

6. Sensors

- **idSensor** : Désigne l'identifiant unique de chaque capteur (clé primaire).
- **idCurrent** : Désigne l'identifiant courant du capteur, puisqu'à chaque remise sous tension l'identifiant change.
- **battery_State** : Donne l'état de la batterie de chaque capteur (1 pour dire qu'il y a de la batterie et 0 pour dire qu'il n'y a plus de batteries).
- **Channel** : Désigne la transmission du capteur correspondant (par exemple : le channel 1 correspond au capteur de température numéro 1, afin de savoir à quel endroit le capteur se trouve).
- **Model** : Désigne le modèle de chaque capteur.

7. Liaisons

- **Sensors** : Chaque capteur est connecté à plusieurs entités de données (température, précipitation, pression et vent). Cela signifie que les capteurs collectent différentes mesures environnementales.
- **Users** : Cette table est utilisée pour gérer des comptes d'accès ou des autorisations donc cette entité n'a pas de liaison directe avec les autres entités du modèle.
- Les tables **Temperature**, **Precipitation**, **Pressure** et **Wind** sont reliées à la table **Sensors** par une relation de type **1,1 → 0,N** :
 - **Un capteur (Sensors)** peut enregistrer de zéro à plusieurs données (**0 à N**) pour chacune des quatre tables.
 - **Chaque donnée** dans ces tables est associée à **exactement un capteur**.



Gantt

Image 26 : "Diagramme de Gantt"

Le diagramme de Gantt présente les tâches réparties entre chacun des étudiants, ce diagramme n'est qu'une représentation globale des tâches. Le diagramme montre les différentes phases du projet, avec des tâches attribuées à chaque acteur. Les barres de chaque tâche sont colorées en fonction de l'acteur responsable, permettant de visualiser les périodes de travail et de coordination entre les acteurs :

1. **Étudiant 1** : Cet acteur est responsable du récepteur. Il va donc en premier lieu créer la base de données, ensuite il va commencer la création du récepteur (réception et traitement des informations). Ensuite il va créer le serveur WebSocket afin de fournir les informations nécessaires au client Android. Enfin il finira par la connexion du récepteur à la base de données.
2. **Étudiant 2** : Cet acteur s'occupe du web. Il va en premier lieu créer les interfaces. Ensuite il va créer établir la connexion entre le contrôleur web et la base données pour commencer le contrôleur météo qui permet d'afficher les données des capteurs. Enfin il s'occupera du contrôleur utilisateur qui permet la connexion des utilisateurs distant au site web.
3. **Étudiant 3** : Cet acteur est chargé de l'application Android. Tout comme l'Étudiant 2 il va commencer par la création des IHM. Il va ensuite créer le client WebSocket permettant de recevoir les données envoyées par le serveur. Il fera ensuite la mise en place des différentes options disponibles aux utilisateurs.